

# Radio Elettronica & Computer

Anno XVIII - N. 6 - Luglio/Agosto 1989 - L. 8.500

Sped. in Abb. Postale Gr. III/70%

Tassa pagata per campione allegato

13 programmi  
per C64  
e C128



**DIDATTICA**  
**LOGO E SUPERBASIC**

**GIOCO**  
**RIMBALZI E TRANELLI**

**UTILITY**  
**MUSICHE PER**  
**OGNI PROGRAMMA**

**ULTRA BASIC**  
**200 PROGRAMMI**  
**IN PIU'**

**AMIGA**  
**NOVITÀ PROVE**  
**SU STRADA**

Gruppo Editoriale  
**JCE**

Trasferimento  
automatico  
dei programmi  
da cassetta a disco

**CASA**  
**Equo canone**  
**automatico**



Il mensile con disco programmi per C64 e C128

# COMMO DISK

Sped. in Abb. Postale Gr. III/70%

Anno III - Luglio/Agosto 1989

N. 33 - L. 13.000

**DRIVE**  
Dominio assoluto  
con quattro tasti

**ARCHIVIAZIONE**  
Programmi  
in ordine  
in un lampo

## BUSINESS GRAPHICS

Grafici a barre,  
a torta, a linee,  
come vuoi

## GIOCO

S.O.S.  
dal naufrago  
dello spazio

## GRAFICA

Conversione  
di schermate

Gruppo Editoriale  
**JCE**



*è in edicola*



# RED HEAT

**- DANKO -**  
L'Est e l'Ovest mettono  
insieme le loro forze per  
combattere un merca-  
nte di droga sovietico.  
Due detective - uno  
russo e uno di Chicago -  
sono spinti da motivi  
diversi, ma la preda è  
una sola.

**- DANKO -**  
Il più caldo "Tie-in" del  
momento, tutto azione e  
grafica superba.





## Rubriche:

Software news pag. 6

Lettere pag. 66

## 12 NE STAMPA DI TUTTI I COLORI!

La stampante: Star LC10 si offre al pubblico a prezzi di grande concorrenza in ben quattro versioni per utenti Amiga o C64. Per gli utenti Amiga, invece, abbiamo provato Antiram, per ottimizzare la gestione della memoria oltre il mezzo mega.

## 16 200 VOLTE PIU' POTENTE



Ultra Basic è una mega espansione multi-purpose che aggiunge al Basic standard ben 200 nuove istruzioni. Le istruzioni messe a disposizione da Ultra Basic riguardano grafica, musica, DOS, debugging e programmazione strutturata. Solo poche altre espansioni in commercio sono altrettanto vaste e potenti.

## 20 LE BASI PER SVILUPPARE IN LOGO



Questo corso di Logo è composto da due parti. La prima, che si conclude con questa puntata, vi insegna a sviluppare programmi Logo discretamente complessi, offrendovi solidissime basi per affrontare la seconda parte del corso che inizieremo al ritorno dalle vacanze estive, sul prossimo numero.

## 29 UNA CARTELLA CLINICA ELETTRONICA

Ogni medico di base, per svolgere seriamente il proprio lavoro, deve costruirsi un archivio contenente la storia sanitaria dei propri assistiti. Queste operazioni possono essere svolte con più efficacia da un programma ottimizzato per questo scopo.

## 33 BOUNDER: PROFESSIONE PALLINA



In questo originale videogame il protagonista è una pallina da tennis che, guidata dal joystick, dovrà superare ostacoli di ogni genere e natura per evitare rimbalzi fatali.

## 36 I RECONDITI PERCHÉ DI AMIGADOS

Concludiamo il discorso sulla strutturazione di un processo multitasking spiegando come avviene la spedizione e la ricezione di dati da una task a un'altra e che cosa si intende per Interruzione, concetto base della programmazione concorrente.

## 42 DI TUTTO UN PO'



I tips di questo mese: cinque utility in linguaggio macchina per dare una marcia in più al C64. Una microespansione per la gestione della bassa risoluzione. Un protettore semplice ed efficiente. Una routine per personalizzare il cursore. Due comodi tool per agevolare chi programma.

## 46 FIAT SONUS! E SUONO FU



Vi siete mai chiesti come il famoso Rob Hubbard componesse le stupende colonne sonore dei migliori videogames? O come certi sofisticati programmi potessero elaborare grafica e musica contemporaneamente? Vorreste potere fare anche voi tutto questo?



# Sommario

## 52 UN MINUTO PER DISTRICARE LA MATASSA



Da oggi si può calcolare il prezzo di qualsiasi affitto abitativo in meno di un minuto, senza possibilità di errore e considerando tutti i parametri che entrano in gioco in particolari situazioni; questo per tutte le zone d'Italia.

## 58 COME PARLARE ALL'INTERPRETE

L'argomento di questa puntata è lo Shell di Qpl, cioè l'interprete di comandi. Questo utilissimo strumento dell'ambiente integrato è molto potente, e mette a disposizione una miriade di comandi con cui praticamente potete fare tutto.

## 62 UN BUON ESEMPIO PER IMPARARE



Il programma che vi mettiamo a disposizione consente di effettuare elaborazioni di vario genere, spaziando da semplici e, per alcuni versi, divertenti calcoli, fino a elaborazioni più complesse di carattere commerciale.

### Caricate così i programmi della cassetta allegata:

Riavvolgete il nastro e digitate **LOAD** seguito da **RETURN** sulla tastiera del C64 e **PLAY** sul registratore. Verrà caricato il programma di presentazione con il menù dei programmi. Digitate **RUN** seguito dalla pressione del tasto **RETURN**. Terminata la presentazione, per caricare uno qualsiasi dei programmi è sufficiente digitare: **LOAD "NOME PROGRAMMA"** seguito dalla pressione del tasto **RETURN**.

**Responsabile Editoriale**  
**Divisione Informatica**  
Marinella Zetti

**Direttore responsabile**  
Paolo Romani

**Caporedattore**  
Fernando Zanini

**Responsabile grafico**  
**Desktop Publishing**  
Adelio Barcella

**Impaginazione elettronica**  
Denise De Matteis

**Segretaria di redazione**  
Alessandra Marini

**Collaboratori**  
Paolo Gussoni, Isa Sestini

**Testi, Programmi, Fotografie e Disegni**  
Riproduzione vietata Copyright.  
Qualsiasi genere di materiale inviato in Redazione, anche se non pubblicato non verrà in nessun caso restituito.

**RadioELETTRONICA & COMPUTER**  
Rivista mensile, una copia L. 8.500, numeri arretrati lire 10.000 cadauno.  
Pubblicazione mensile registrata presso il Tribunale di Monza n. 679 del 28/11/88.

**Fotolito:** Bassoli - Milano.  
**Stampa:** GEMM Grafica srl, Paderno Dugnano (MI).  
**Diffusione:** Concessionario esclusivo per l'Italia A.&G. Marco SpA, Via Fortezza 27 - 20126 Milano. Spedizione in abb. post. gruppo III/70.

**Abbonamenti:** Annuale L. 54.000, estero L. 83.000. Biennale L. 97.500

RadioELETTRONICA & COMPUTER è titolare in esclusiva per l'Italia dei testi e dei progetti di Radio Plans e Electronique Pratique, periodici del gruppo Société Parisienne d'Édition.

Gruppo Editoriale  
**JCE**

**Gruppo Editoriale JCE srl**  
Sede legale, Direzione, Redazione, Amministrazione  
Via Ferri 6 - 20092 Cinisello Balsamo (MI)  
Tel. 02/66025.1 - Telex 352376 JCE MIL I  
Telefax 61.27.620

**Direzione Amministrativa:** Walter Buzzavo

**Direttore Commerciale:** Giuseppe Tiani

**Pubblicità e Marketing**  
Gruppo Editoriale JCE - Divisione Pubblicità  
Via Ferri 6 - 20092 Cinisello Balsamo (MI)  
Tel. 02/66025.1

**Concessionario esclusivo per Roma, Lazio e centro sud:**  
UNION MEDIA srl - Via C. Fracassini, 18  
00198 Roma - Tel. 06/3215434 (13 linee R.A.)  
Telex 630206 UNION I - Telefax 06/3215678

**Abbonamenti:** Le richieste di informazioni sugli abbonamenti in corso si ricevono per telefono tutti i giorni lavorativi dalle ore 9 alle 12. Tel. 02/66025311- 66025338

**Spedizioni:** Daniela Radicchi

I versamenti vanno indirizzati a:  
Gruppo Editoriale JCE srl, Via Ferri 6  
20092 Cinisello Balsamo (MI), mediante l'emissione di assegno circolare, cartolina vaglia o utilizzando il c.c.p. n. 351205. Per i cambi di indirizzo allegare alla comunicazione l'importo di L. 3.000, anche in francobolli, e indicare insieme al nuovo anche il vecchio indirizzo.

Associato al

**CST** Consorzio  
Stampa  
Specializzata  
Tecnica

Testata in corso di certificazione  
obbligatoria secondo quanto stabilito  
dal Regolamento del C.S.S.T.



Mensile associato  
all'USPI  
Unione Stampa  
Periodica Italiana



## Photon Paint

Questo mese iniziamo la rubrica sulle novità software con un programma per Amiga davvero strabiliante: Photon Paint in italiano, un programma grafico dotato di enormi possibilità di sfruttamento: l'unica cosa che non può fare è disegnare da solo.

Photon Paint è un programma grafico artistico, nel senso che non serve per eseguire disegni tecnici, ma è un prezioso alleato dei grafici. La sua caratteristica principale è la maneggevolezza nel trattamento dei colori, infatti Photon Paint funziona in modo Hold and Modify, che in seguito chiameremo semplicemente HAM seguendo la terminologia corrente.

Se desiderate avere una dettagliata descrizione di che cosa significa modo HAM



*Ecco che cosa è possibile realizzare con le eccezionali capacità di Photon Paint.*

vi consigliamo di leggere la rubrica Accessori di *RadioElettronica & Computer* di giugno, dove l'argomento è trattato in modo particolareggiato nel corso della descrizione di Easy View. In ogni modo è più che sufficiente sapere che il modo HAM consente la raffigurazione sullo schermo di Amiga di tutti i 4096 colori di cui è capace. Naturalmente HAM comporta alcune limitazioni al fine di ottimizzare l'occupazione della memoria: nel caso di Photon Paint queste limitazioni talvolta comportano un certo rallentamento nelle operazioni che gestiscono un gran numero di pixel contemporaneamente come l'opzione Fill.

Photon Paint è un editor grafico, ovvero è un programma che consente sia la modifica disegni creati con altri programmi come l'ottimo Sculpt 3D o digitalizzatori video, sia la creazione di disegni ex novo. Non esistono problemi di incompatibilità con gli altri programmi, infatti Photon Paint utilizza il praticissimo sistema IFF che è lo standard indiscusso per i file grafici di Amiga. Photon Paint è capace anche di caricare file grafici differenti: in questo caso prima di procedere al caricamento informa l'utente che qualcosa potrebbe non funzionare poiché non riconosce lo standard.

Photon Paint è distribuito in una bella confezione che comprende il disco programma, un disco dati che raccoglie molte immagini di notevole effetto e il manuale. Sia il manuale che il programma sono tradotti in italiano e la traduzione di entrambe le parti del pacchetto software è buona; è molto utile disporre del manuale in italiano poiché il programma non può essere sfruttato a fondo senza l'ausilio delle istruzioni.

Vediamo ora che cosa può fare Photon Paint. Iniziamo dalle opzioni semplici: è possibile disegnare a mano libera in tempo reale con qualsiasi colore scelto nella palette di 4096. La cosa non è assolutamente banale poiché il modo HAM, per consentire l'uso di 4096 colori contemporaneamente, implica delle limitazioni sulla scelta del colore dei pixel adiacenti. Photon Paint gestisce nel migliore dei modi le complicazioni introdotte dal modo HAM. È possibile realizzare automaticamente le più comuni figure geometriche, ovvero rettangoli, cerchi ed ellissi. Tali figure possono essere vuote, ossia si disegna il solo contorno, oppure piene. C'è la comunissima operazione Fill, che però è particolarmente sofisticata in quanto presenta tre diverse possibilità: Riempimento a Inondazione, ovvero esteso a tutti i punti adiacenti di un'area scelta, Riempimento Confinato, che considera l'area circondata da un particolare colore e infine Forma libera, che consente di disegnare a mano libera un'area qualsiasi da riempire. Naturalmente l'area può essere coperta da un colore uniforme o da un fondo selezionabile con un apposi-



to comando. È possibile disegnare servendosi di pennelli (brushes per chi è abituato alla terminologia inglese). I pennelli possono essere ruotati, invertiti, inclinati con effetti prospettici, ingranditi o rimpiccioliti. Se la memoria disponibile lo consente Photon Paint gestisce contemporaneamente due disegni selezionabili semplicemente premendo un tasto, oltre allo schermo di Undo che consente l'annullamento dell'ultima operazione compiuta.

Tutte queste operazioni sono ormai comuni a qualsiasi programma grafico, a parte la gestione del modo HAM naturalmente. Ma Photon Paint può fare molto di più. Il menù di selezione è una finestra video che può essere spostata ovunque nello schermo in modo da scoprire la parte interessata del disegno, inoltre può essere rimpicciolito o addirittura eliminato. È possibile sfruttare i pennelli in modo molto sofisticato e nello stesso tempo molto semplice: si possono effettuare torsioni, trasparenze e soprattutto è possibile avvolgere i pennelli attorno a solidi tridimensionali. Quest'ultima operazione è particolarmente sensazionale in quanto permette anche di illuminare il solido risultante da una direzione a piacere: in sostanza consente di ottenere eccellenti risultati con un minimo sforzo. Photon Paint dispone di una funzione Zoom molto efficace che permette la selezione della dimensione dei pixel e il facile spostamento della zona ingrandita. Non manca un completo menù per la gestione delle scritte all'interno dei disegni che permette la gestione di tutti i font contenuti nell'apposita directory del disco di sistema (quello con cui si effettua il boot) oppure in un'altra directory facendo uso dell'istruzione Assign di AmigaDos.

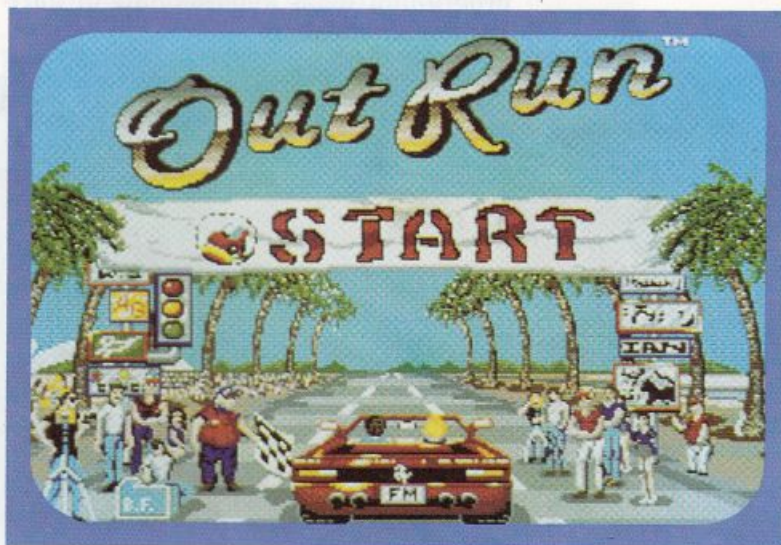
Per concludere vediamo le opzioni per la selezione dei colori: il colore di fondo può essere trasparente o opaco, inoltre è sempre disponibile una palette con 64 colori modificabili a piacimento senza alterare il disegno già realizzato. Per impostare un colore nuovo si può scegliere tra il sistema RGB, ovvero la scelta della dose di rosso, blu e verde, e il sistema HSV (Hue Saturation Value). Per gli esperti è anche possibile selezionare i 16 colori di base del sistema HAM, al fine di ottene-

re i risultati più efficaci nelle sfumature.

Siamo convinti che quanto vi abbiamo detto è più che sufficiente per convincervi che Photon Paint è davvero un programma completo e realizzato con molta attenzione e competenza. Ci sarebbe ancora molto da dire per descrivere tutte le operazioni che possono essere compiute con questo eccellente programma, pertanto non possiamo che concludere la prova di Photon Paint esprimendo la massima soddisfazione per questo prodotto della Microillusions.

### Giochi per Amiga

Iniziamo con le versioni per Amiga di due famosissimi giochi da bar creati dalla Sega Enterprises. Si tratta di Out Run



e di Super Hang On che, come tutti gli appassionati sanno, sono rispettivamente un simulatore di gara automobilistica e motociclistica.

#### • Out Run.

Con Out Run vi trovate in compagnia di una fantastica bionda alla guida di una Ferrari (una prospettiva allettante) e dovete percorrere una bella strada, che in molti tratti costeggia il mare, entro un tempo assegnato. Secondo uno schema comune a tutti i giochi di questo tipo se raggiungete il traguardo vi viene assegnato un tempo supplementare in modo da tentare il percorso successivo.

Un giudizio su Out Run è a nostro av-

*L'attraente schermata introduttiva di Out Run. La grafica del gioco non è da meno.*



viso molto impegnativo: certamente la versione originale rappresenta uno dei migliori videogame che abbia calcato la scena negli ultimi anni, tuttavia la versione per Amiga creata dalla US Gold ci ha lasciato qualche perplessità. La grafica è di buona qualità, ma siamo certi che con Amiga si sarebbe potuto ottenere un risultato più simile all'originale, inoltre la manovrabilità della macchina non ci ha lasciato pienamente soddisfatti. Il sonoro invece è davvero ottimo. Comunque, per i più esigenti, la US Gold ha dotato il programma di una cassetta audio con la colonna sonora del gioco.

Resta da dire che, pur con qualche difettuccio, Out Run è un videogioco molto piacevole e appassionante che sicuramente non può mancare dalla raccolta di programmi degli appassionati di gare automobilistiche.



*Super Hang On.  
La griglia  
di partenza  
un attimo  
prima dello start.*

### • Super Hang On.

La Electric Dream Software ha eseguito un lavoro più accurato con Super Hang On. La struttura del gioco è sostanzialmente identica a Out Run, pertanto ci limitiamo a elencarvi le piccole differenze. Come abbiamo già accennato si tratta di una gara motociclistica, quello che non vi avevamo ancora detto, ma che voi probabilmente già sapete, è che vi trovate alla guida di una moto dotata di un incredibile controllo del turbo-compressore, simile alla "manetta" usata l'anno scorso sulle vetture di Formula 1.

Quando raggiungete la velocità massima (280 km/h) avete la possibilità di inserire il turbo ed eseguire un'ulteriore poderosa accelerazione. Inutile dire che il controllo della moto è reso estremamente difficoltoso dall'inserimento del turbo, d'altra parte se si desidera ottenere un buon risultato è necessario imparare a servirsene.

La dinamica di gioco di Super Hang On è eccellente: si tratta di un ottimo simulatore di guida motociclistica. L'unica cosa che ci lascia perplessi è la strana tenuta di strada esibita dalla moto: è facilissimo controllarla, anche a velocità elevatissima a patto ovviamente di evitare la collisione con le altre moto, se ci si trova all'interno della curva, mentre si perde completamente l'aderenza se si imposta una traiettoria troppo larga. Questo comportamento è sicuramente molto dubbio poiché fino a prova contraria più si stringe una curva, più è necessario rallentare.

Una caratteristica degna di nota invece è data dalla possibilità di controllare la moto sia con il joystick sia con il mouse, oltre che con la tastiera. Particolarmente pregevole è inoltre il controllo di sensibilità sia per il mouse sia per il joystick, che nel primo caso consente la regolazione dell'escursione massima del mouse, mentre nel secondo caso consente di variare la velocità con cui la moto si piega.

Considerando anche la buona qualità della grafica e del sonoro possiamo affermare che Super Hang On è un eccellente prodotto, a nostro avviso migliore di Out Run.

### • The Real Ghostbusters.

Cambiamo completamente genere, lasciando il mondo delle corse per entrare nel mondo della fantasia più sfrenata. È di scena The Real Ghostbuster della Activision. La struttura portante del gioco è molto banale: in poche parole The Real Ghostbuster è il classico multischermo in cui si deve girare in molti posti e si deve sparare a tutti i nemici. Nulla più. In realtà, anche se la dinamica del gioco è piuttosto misera, dobbiamo dire che è stata arricchita con molta abilità, infatti i mostri che dovete uccidere e tra-



sformare in simpatici fantasmini sono davvero molto vari e variopinti. Lo scenario di gioco è altrettanto colorato e ricco e anche le due armi che avete a disposizione sono decisamente scenografiche.

Non poteva mancare in un gioco di questo genere un eccellente scrolling in ben otto direzioni. Non potevano mancare sofisticati effetti sonori per completare il quadro, nonché un'introduzione con musica originale tratta dall'omonimo file.

Per una volta non abbiamo alcuna difficoltà a dare un giudizio al programma in prova: si tratta di un videogioco sviluppato su un'idea piuttosto banale, ma realizzato in modo impeccabile.

#### • Pac-land.

Terminiamo con un breve commento a Pac-land della Quicksilva. Anche in questo caso abbiamo a che fare con la conversione di un famoso coin-up, il Pac-land della Namco. La conversione è abbastanza fedele, tuttavia il programma non ci ha entusiasmato poiché il gioco è davvero banale e, a differenza di *The Real Ghostbusters*, non è neanche dotato di grafica o sonoro di ottima qualità. Siamo certi che Pac-land possa piacere ai più giovani utenti di Amiga, anche se a nostro avviso il successo di questo gioco è da attribuire in grande misura alla presenza del mitico Pac-man.

#### Giochi per C 64

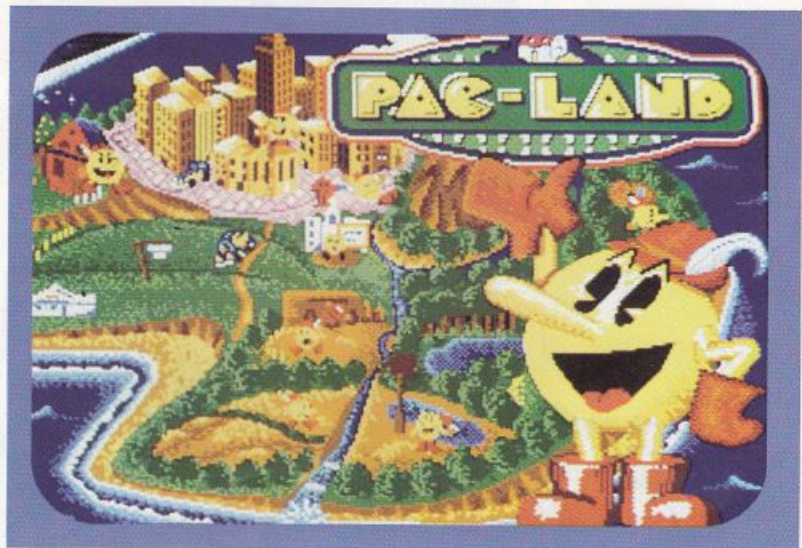
La nostra rassegna mensile delle novità software per C64 distribuite di recente dalla Leader presenta questa volta dei programmi che si possono inserire in due specifiche categorie; accontenteremo coloro i quali preferiscono i giochi di avventura e strategia ma anche il pubblico che predilige cimentarsi con il joystick in frenetici videogame d'azione.

#### • Hillsfar.

Questo programma è una delle ultime produzioni che la Strategic Simulations Inc. ha creato per la serie *Forgotten Realms*. Abbiamo già parlato molto bene di *Battles of Napoleon*, sempre della stessa casa, sottolineando l'estrema cura con cui è stato trattato ogni particolare di un gioco già di per se stesso ben articolato.

Analogo commento si può fare per il programma ora in esame; pur non essendo ispirato a vicende storiche bensì a leggende medievali l'avventura di cui il gioco tratta si sviluppa in uno scenario alquanto realistico.

Hillsfar è un villaggio governato da un potente despota, le sue temute guardie mercenarie mantengono l'ordine controllando ogni angolo del paese alla ricerca di chiunque abbia intenzione di sovvertire il potere. Voi non avete una missione precisa ma vedrete che non sarà difficile ritrovarvi nel bel mezzo di un'appassionante avventura: è sufficiente oltrepassare le porte del villaggio e addentrarvi nei vicoli e negli edifici; qui particolari situazioni e strani incontri non vi renderanno molto facile la vita. Per uscire dalle intricate vicende in cui sarete coinvol-



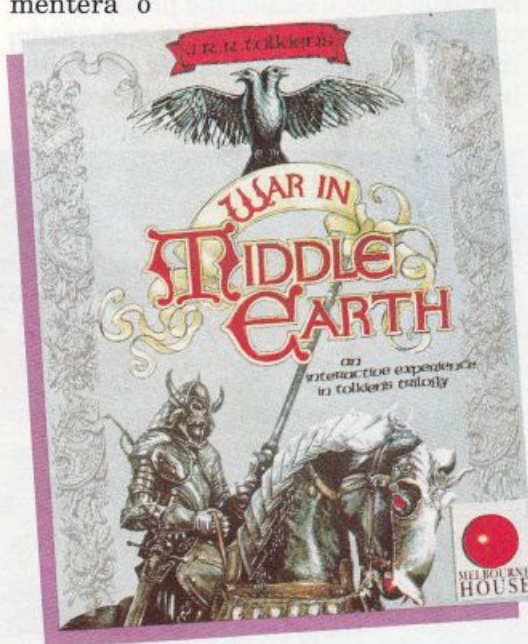
ti dovrete fare affidamento non solo alla vostra furbizia e saggezza ma anche alle capacità proprie di un avventuriero che si rispetti: saper cavalcare, cavarsela con l'arco e le frecce, saper combattere in corpo a corpo all'ultimo sangue, essere abili in piccoli trucchi di destrezza come scassinare una serratura e possedere molte altre doti definibili come l'arte di arrangiarsi.

Un'opportunità positiva del gioco è quella di potervi creare in tutti i suoi aspetti il personaggio che vi rappresenterà nello scenario di Hillsfar. All'inizio del gioco infatti potrete caricare un per-

*Pac-land .  
La coloratissima  
schermata  
introduttiva  
del programma.*



sonaggio già creato e salvato su disco (magari ripreso direttamente da un altro programma della serie *Forgotten Realms*), oppure plasmarlo a vostro piacimento determinandone il sesso, la razza (umano, elfo, gnomo), la classe (prete, lottatore, mago, ladro) e addirittura la filosofia di vita. Una volta fatte le scelte il computer genererà automaticamente dei parametri che evidenzieranno le particolari doti del personaggio da voi creato; essi riguardano forza, intelligenza, abilità nelle arti magiche, destrezza, salute e carisma. La somma di queste qualità determina il punteggio di partenza che, con il procedere del gioco, aumenterà o



*War In The Middle Earth. L'immagine sull'attraente confezione del gioco.*

diminuirà a seconda dell'andamento dell'avventura; quando si annulla il personaggio muore. Un altro punteggio, denominato Punteggio d'esperienza, cresce con i tesori accumulati e con le vicende superate positivamente ed è indice del buon esito dell'avventura.

Hillsfar non è sicuramente uno di quei giochi il cui divertimento si esaurisce in poco tempo: ogni volta si incomincia un'avventura diversa poiché le possibilità di azione e la varietà di situazioni sono talmente tante che è quasi impossibile il ripetersi di una stessa vicenda. Non trascurabile l'idea di poter personalizzare il personaggio e farlo vivere non solo ad

Hillsfar ma in tutti gli altri scenari della stessa serie di action adventures. Molto buona la grafica sia nella parte essenzialmente statica (l'avventure vero e proprio), sia nella parte più prettamente d'azione.

### • War In The Middle Earth.

Rimaniamo nel genere presentando un gioco della Melbourn House ispirato nientemeno che ai celebri racconti di J. R. R. Tolkien, l'autore di *Il signore degli anelli* e di *Lo Hobbit*. Da tale penna non poteva che uscire un'ottima sceneggiatura per il videogame in questione; non ci dilungheremo nel descrivere il fantastico mondo in cui la vicenda si svolge, chi ha letto qualcosa di Tolkien saprà già di cosa si tratta, gli altri rimarranno piacevolmente sorpresi nello scoprire i personaggi e le epiche vicende che animano lo scenario del gioco.

Frodo è uno hobbit (mite creatura delle campagne) che ha ereditato un anello dai poteri soprannaturali: il suo possessore non solo può diventare invisibile ma assume una potenza malefica che lo rende capace di soggiogare l'intera Middle Earth, l'immaginaria regione scenario del gioco.

Il vostro compito è quello di gettare l'anello nel Vulcano della morte, l'unico modo per distruggere il malefico strumento affinché non cada nelle mani di Sauron, il maligno signore delle tenebre. Il problema consiste nel fatto che il cratere si trova proprio nel bel mezzo del regno di Sauron e che quindi prima di accedervi sarà necessario fare i conti con le milizie del tiranno. Frodo, comunque, non è da solo in balia di un intero esercito: a fargli compagnia e ad aiutarlo nello sgattaiolare fra le file nemiche vi sono otto fidati amici e tutti gli alleati del popolo degli Hobbits. Sta a voi guidarli tutti e disporli nel territorio di Middle Earth in modo da favorire il viaggio dell'anello. Potrete fare ciò attraverso specifici comandi di facile attuazione come scegliere la destinazione di un certo gruppo di uomini, ordinare a più alleati di unirsi, fare in modo che una certa compagnia segua un'altra e così via. Facendo scorrere il cursore sulla mappa di Middle Earth si incontrano le truppe alleate contrassegnate da



uno scudo: premendo il tasto Fire del joystick in corrispondenza di una di esse appaiono tutte le informazioni che la riguardano come il nome, il numero di componenti, il nome del comandante, la destinazione, l'energia a disposizione, la determinazione, la virtù, il coraggio e la forza. Questi fattori determinano la vittoria o la sconfitta nel caso che la compagnia venga a trovarsi di fronte alle truppe nemiche. L'anello, che permette a chi lo indossa di rendersi invisibile, può essere anche passato di mano, ovviamente ad un alleato, nel caso questi abbia maggior probabilità di arrivare sano e salvo al vulcano. Le battaglie tra fazioni rivali possono essere combattute sia autonomamente dal computer sia con il vostro diretto intervento tramite joystick; in quest'ultimo caso il campo di battaglia viene zoomato ed è possibile controllare il personaggio segnato dal cursore.

In definitiva War In The Middle Earth è sia un wargame di strategia e inventiva sia un gioco d'azione e destrezza, grazie appunto alla possibilità di interagire direttamente con i personaggi del gioco. Il tutto è completato da un affascinante scenario che fa da ottimo sfondo a un'avventura così ben congegnata. Purtroppo la musica e la grafica sono un po' al di sotto della qualità generale del programma.

#### • Blasteroids.

Accontentiamo ora gli amanti dei giochi ad alta velocità introducendo Blasteroids della Image. Il nome fa subito pensare a un glorioso precursore degli odierni videogame, Asteroid, e in effetti questo programma non è altro che una sua rielaborazione. Per i più giovani ricordiamo che in Asteroid una navicella vagava per lo spazio interstellare cercando di schivare o di colpire una marea di asteroidi che, sotto l'effetto dei laser, si frantumavano in pezzi sempre più piccoli, fino a scomparire. Il punteggio si incrementava in base agli asteroidi distrutti e alle schermate completate.

Lo schema base del gioco è rispettato anche in questa nuova versione ma il tutto è arricchito da alcune trovate che mirano ad aumentarne la giocabilità. Infatti il difetto principale di Asteroid era l'estrema ripetitività dell'azione. In Blaste-

roids si sono introdotti elementi nuovi che in parte hanno sopperito ai difetti del predecessore. Innanzitutto vi è la possibilità di scegliere fra tre diversi tipi di navicella, ognuna con una caratteristica peculiare (velocità, potenza di fuoco, resistenza); altra novità consiste nella diversificazione degli asteroidi, ognuno di essi ha un comportamento ben preciso: alcuni non possono essere distrutti ma solo arrestati, altri, una volta colpiti, fanno scaturire una pillola di energia che, se intercettata, aumenterà l'autonomia della navicella. Inoltre, insieme a voi, vagano nello spazio sia altri oggetti volanti nemici che cercheranno di colpirvi con il loro armamento o di urtarvi, sia armi e strumenti particolari che agevoleranno il vostro compito.

Alla fine di ogni schermata si entra in un quadro particolare per la scelta del livello successivo. Le possibili galassie dove continuare il gioco sono simbolicamente raffigurate corredate delle insidie in esse contenute. In questo modo il giocatore può valutare la difficoltà e decidere di conseguenza se entrare o meno in un dato settore. Dopo aver completato tutte le galassie l'esito positivo del gioco è determinato da un'ultima prova, lo scontro finale con un essere mostruoso dagli infiniti tentacoli. Se lo sconfiggerete avrete superato l'intero percorso.

Anche se non spicca per originalità, Blasteroids è un buon arcade che deve molta della sua giocabilità all'immediatezza e alla semplicità dell'azione. Apprezzabili alcune trovate scenografiche come la modalità della scelta iniziale del grado di difficoltà, effettuabile facendo attirare la navicella da quattro spirali magnetiche rappresentanti i vari livelli, e la scelta della galassia in cui continuare il gioco. Efficace inoltre l'idea di determinare la distruzione della navicella non con una semplice collisione, come accadeva per Asteroid, ma con l'esaurimento dell'energia dovuto sia a collisioni che ad accelerazioni del mezzo. In questo modo il giocatore non si deve limitare a schivare gli asteroidi ma deve anche saper amministrare il proprio velivolo. Discreti la grafica e il sottofondo sonoro.

**Gianni Arioli  
Massimiliano Del Rio**





**C64**

## ACCESSORI

*Ecco la stampante del momento: Star LC10 si offre al pubblico a prezzi di grande concorrenza in ben quattro versioni per utenti Amiga o C64. Esclusivamente per gli utenti Amiga, invece, abbiamo provato Antiram, per ottimizzare la gestione della memoria oltre il mezzo mega.*

# Ne stampa di tutti i colori!



Questo mese la rubrica accessori è quasi interamente dedicata a un ottimo strumento utilissimo ai proprietari sia di C64 sia di Amiga: una stampante.

### Star LC10

Esistono quattro diverse versioni di questa stampante che possono soddisfare qualunque utente: ci sono due versioni per C64, una in bianco e nero e l'altra a colori, e due versioni dotate di porta parallela che possono essere utilizzate con Amiga o con qualunque IBM compatibile dotato di uscita Centronics.

Le versioni per C64 sono dotate di un'apposita interfaccia che permette il collegamento direttamente al bus seriale del computer, pertanto non esiste alcun problema di allacciamento o di compatibilità. Le versioni dotate di porta parallela invece possono essere collegate senza alcun problema a qualunque porta Centronics: nel caso si utilizzizzi un Amiga, per ottenere la corretta gestione di tutti i caratteri di controllo del colore e dei vari effetti (sottolineato, grassetto, corsivo eccetera) è necessario





C64E

settare dal menù Preferences l'opzione Epson X, poiché la Star LC10 emula il funzionamento della Epson LX-800. Tramite dip switch è anche possibile ottenere il funzionamento della Star LC10 in emulazione della IBM ProPrinter II.

Noi abbiamo provato la versione a colori per Amiga, con la quale abbiamo stampato i disegni che potete ammirare sulle pagine della rivista, ma vi possiamo assicurare che le altre versioni della stampante sono assolutamente identiche, tranne naturalmente per quanto riguarda la stampa a colori. Il nastro della stampante a colori è a quattro bande di colore nero, blu, rosso e giallo, ma quando si preferisce stampare sono in bianco e nero è possibile utilizzare un nastro nero, più economico. Il nastro a colori consente di ottenere la stampa con i sette colori fondamentali, ossia i quattro di base e altri tre composti (verde, viola e arancione). Questi sette colori sono ottenibili direttamente ma naturalmente, se disponete di un software che ne gestisce un numero maggiore, potete ottenere numerosissime sfumature come per i disegni in figura.

Un primo commento molto positivo viene dalla lettura del manuale operativo: non solo nella confezione è possibile trovare il manuale in italiano oltre alla versione originale in inglese, ma è inoltre stato realizzato in modo molto completo e preciso. Ogni operazione eseguibile con la Star LC10 è descritta in modo semplice e particolareggiato, inoltre non manca un capitolo dedicato alla gestione della stampante via Basic o Dos, né mancano istruzioni sull'uso di LC10 con il software commerciale, pertanto riteniamo che nessuno dovrebbe incontrare difficoltà a sfruttare tutte le numerose possibilità offerte dalla stampante.

La Star LC10 è una stampante a impatto a nove aghi. Si tratta di uno standard ormai più che affermato che, tramite alcuni accorgimenti quali la doppia battitura, consente di ottenere una stampa di buona qualità. Sono disponibili tre

## Star LC10 COLOUR

Sono disponibili diversi *Fonts*

Courier	Abcdefghijklmnopqrstuvwxyz	1234567890
Sanserif	Abcdefghijklmnopqrstuvwxyz	1234567890
ORATOR	ABCDEFGHIJKLMNopqrstuvwxyz	1234567890
Draft	Abcdefghijklmnopqrstuvwxyz	1234567890

Doppia larghezza oppure Doppia altezza oppure  
Doppia larghezza e altezza.

Figura 1.

caratteri di stampa, NLQ (Near Letter Quality) e un font draft che consente la stampa molto veloce di tabulati che non richiedono una grande qualità. In **tavola 1** potete ammirare le prove di stampa con i diversi font: certamente non si raggiunge la qualità tipica delle stampanti a 24 aghi, tuttavia i tre font NLQ consentono di raggiungere buoni risultati.

Come tutte le stampanti di buona qualità sono disponibili per ogni font il modo italico, il sotto e sopra lineato, il grassetto, gli esponenti e i deponenti. Anche la larghezza dei caratteri è molto flessibile: si può utilizzare il modo Pica (10 caratteri per pollice) che consente le comunissime 80 colonne, oppure il modo Elite (12 caratteri per pollice). Entrambi gli stili possono essere condensati: in particolare con il modo Elite compresso si raggiungono 160 colonne. Si può utilizzare anche il modo proporzionale che assegna a ciascun carattere lo spazio necessario, ovvero per esempio poco per la i e la l e molto per la m e la w. Il modo proporzionale consente di raggiungere i risultati migliori in termini di qualità di stampa, ma deve essere gestito bene dal software se si desidera la stampa formattata rispetto a entrambi i margini in quanto il numero di caratteri per riga è variabile.

LC10 consente anche la stampa con caratteri di larghezza e/o di altezza doppia o addirittura quadrupla. La stampante provvede automaticamente a crearsi lo spazio necessario per l'inserimento dei caratteri in doppia altezza, tuttavia è





**C64**

## ACCESSORI

necessario prestare attenzione quando si usa un word processor che formatta la pagina secondo un numero fisso di linee poiché è molto facile uscire dalla lunghezza standard della pagina.

Il controllo delle opzioni di stampa più utilizzate è molto flessibile: è possibile usare appositi codici (istruzione CHR\$ in Basic) per settare qualunque modo di stampa. In questo modo si mantiene la completa compatibilità con lo standard Epson. Se però si desidera lavorare più velocemente, la versione a colori di LC10 possiede un sistema a doppia parentesi che consente l'inserimento di molte opzioni in modo semplicissimo. Se per esempio volete modificare il colore di stampa potete inviare i codici 27, 114 più il numero del colore prescelto, oppure potete semplicemente scrivere in mezzo al vostro testo ((C))0 per passare al nero, ((C))1 per passare al rosso e così via per i sette colori fondamentali. Analogamente per cambiare il font di stampa esiste un'apposita sequenza di codici, ma è sufficiente scrivere ((F))n dove n è un numero che varia da 0 a 3. Anche in questo caso però si paga la semplicità d'uso con i problemi di formattazione delle linee di testo, infatti un word processor non può sapere che la scritta ((F))2 è una sequenza di controllo piuttosto che una parola.

Una caratteristica a nostro avviso molto utile di questa stampante è la possibilità di utilizzare sia i fogli singoli sia i moduli continui in modo molto flessibile. LC10 dispone di due sistemi di trazione: a rullo di gomma e a trattore. In questo modo è possibile gestire nel modo migliore entrambi i supporti cartacei, ma la caratteristica che porta LC10 un gradino più in alto di molte altre stampanti della stessa categoria consiste nella possibilità di stampare su foglio singolo senza sganciare il foglio continuo. È sufficiente premere due tasti del pannello di controllo e spostare la leva di commutazione tra il trattore e il rullo per fare arretrare il modulo continuo e inserire il foglio singolo. L'inserimento di quest'ultimo inoltre è molto agevole in quanto LC10 dispone di serie di un accessorio che ne permette l'immissione semiautomatica. Per tornare a utilizzare il modulo continuo basta estrarre il foglio singo-

lo e premere un apposito tasto per riportare la stampante on line.

Passiamo ora a esaminare la gestione della grafica. LC10 consente la definizione da parte dell'utente di ben 192 caratteri draft oppure 78 caratteri NLQ. Questo significa che oltre ai vari caratteri alfanumerici e grafici disponibili di default è possibile programmare molti altri simboli da utilizzare in svariati contesti. Questi simboli grafici possono essere utilizzati per stampare un logo, oppure per stampare le lettere greche o dei simboli matematici. L'operazione di definizione è un po' laboriosa, tuttavia ad essa è dedicato un intero capitolo del manuale, e ogni particolare è accuratamente descritto. Non manca ovviamente la stampa grafica a matrice di punti. Sono possibili sette diverse risoluzioni, da un minimo di 60 punti per pollice a un massimo di 240 punti. Il modo a maggiore densità consente di raggiungere risultati molto buoni, che comunque dipendono in gran parte dalla bontà del software di gestione. Anche a questo riguardo comunque non possiamo che ribadire la completezza del manuale che dovrebbe consentire davvero a tutti l'uso di procedure anche sofisticate come la stampa grafica.

Il nostro giudizio sulla Star LC10 non può che essere positivo: si tratta di una stampante piuttosto economica che consente di ottenere risultati di una categoria superiore, sempre che non si pretenda la qualità dei 24 aghi o una stampa a colori raffinata. La velocità è buona durante la stampa alfanumerica, in questo caso l'uso dei colori non rallenta in modo molto vistoso. La stampa grafica in nero costringe la LC10 a un notevole rallentamento, mentre la stampa grafica a colori è estremamente lenta.

La Star LC10 è sicuramente la stampante ideale per molti hobbisti, anche nella versione a colori per chi desidera qualche prestazione in più. Non consiglieremmo invece la LC10 a chi desideri sfruttarne intensamente le capacità di stampa a colori poiché, oltre alla notevole lentezza, si deve tenere presente che la definizione della stampa a colori è decisamente inferiore alla stampa in nero.

Potete trovare la Star LC10 presso qualunque buon rivenditore di hardwa-





C64



re. In particolare è disponibile un'ottima offerta presso la Niwa di via Buozzi 94 a Sesto S.Giovanni (MI), tel. (02)2620312.

### Antiram

Per concludere questa rubrica vediamo un piccolo e semplicissimo accessorio prezioso per tutti gli utenti di Amiga 2000 o di Amiga 500 dotata di espansione di memoria RAM espansa: Antiram.

Si tratta di un dispositivo che consente di disattivare la Fast Ram di Amiga semplicemente agendo su un piccolo interruttore. La memoria RAM di Amiga è suddivisa in due parti: la Chip RAM che occupa i primi 512 Kb e la Fast RAM che copre la parte restante. La Chip RAM gioca un ruolo particolare in Amiga poiché è l'unica zona di memoria cui possono accedere i chip custom, il che significa per esempio che la memoria video è sempre contenuta nella Chip RAM.

La Fast RAM viceversa presenta un vantaggio in quanto consente un più rapido accesso del microprocessore e, per questo motivo, esiste il programmino FastMemFirst che impone ad Amiga di utilizzare la Fast RAM prima della Chip RAM. Amiga 2000 esce di fabbrica dotata di 512 Kb di Chip RAM e 512 Kb di Fast RAM, mentre Amiga 500 dispone della sola Chip RAM, ma moltissimi utenti hanno acquistato l'espansione a 1 mega.

Purtroppo però alcuni programmi non accettano la presenza della Fast RAM:

potrebbe apparire un controsenso, ma questo accade talvolta a causa delle protezioni anti-copia. Il sistema operativo di Amiga, o meglio le ultime versioni di Workbench, è dotato di un programma che elimina via software la Fast RAM, ma in certi casi è necessario operare via hardware. Poiché nessuno si sogna di smontare Amiga per inserire e levare l'espansione di memoria, l'unica soluzione è Antiram. L'installazione di Antiram è un'operazione molto semplice. Si apre il computer prestando attenzione alla connessione della tastiera nel caso di Amiga 500, si cerca Gary (un chip custom facilmente riconoscibile dalla sigla indicata nelle istruzioni), lo si estrae dallo zoccolo usando moltissima attenzione e delicatezza, si inserisce al suo posto lo zoccolo di Antiram, si rimette a posto Gary e per finire si sistema il deviatore di Antiram all'esterno del computer. L'intera operazione richiede pochi minuti, ma deve essere eseguita prestando molta attenzione alla posizione dei vari circuiti. A questo punto, con il deviatore in una posizione Amiga, l'intera memoria di Amiga è a disposizione, mentre spostando il deviatore la sola Chip RAM resta a disposizione, com'è facile verificare con il comando AmigaDos Avail che mostra le risorse di RAM della macchina.

Antiram è prodotto dalla ditta Tesi di Milano ed è anch'esso disponibile presso la Niwa.

**Gianni Arioli**





C64

## ESPANSIONE

*Ultra Basic è una mega espansione multipurpose che aggiunge al Basic standard ben 200 nuove istruzioni. Le istruzioni messe a disposizione da Ultra Basic riguardano grafica, musica, DOS, debugging e programmazione strutturata. Solo poche altre espansioni in commercio sono altrettanto vaste e soprattutto altrettanto potenti.*

# 200 volte più potente

### Prima puntata

L'espansione di questo mese è una delle più vaste e potenti in circolazione. Ultra Basic, questo è il suo nome, aggiunge all'interprete standard qualcosa come 200 nuove istruzioni che spaziano dalla grafica alla musica, dalla programmazione strutturata al debug dei programmi. Ultra Basic è quindi una mega espansione nel vero senso della parola, perché ha proprio tutto ciò di cui un programmatore ha bisogno per re-

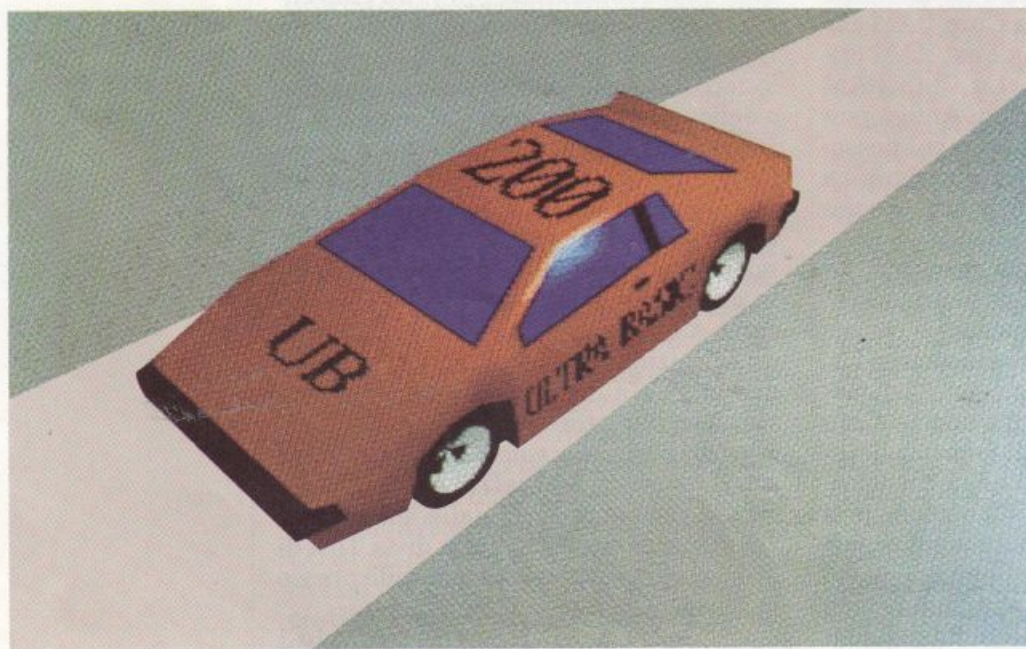
alizzare software senza troppi sforzi.

Pur essendo assolutamente generale, questa espansione ha due punti di forza: le istruzioni grafiche e quelle di controllo. Le istruzioni grafiche permettono di gestire con estrema facilità ogni singolo aspetto della grafica in bassa e alta risoluzione, compresi sprite e caratteri programmabili. Tra le istruzioni grafiche sono sicuramente da segnalare quelle che permettono di muovere gli sprite da

interruzione e di copiarli nella pagina grafica in un punto determinato. Per quanto riguarda le istruzioni di controllo, praticamente non si può fare un estratto di quelle più interessanti perché sono tutte molto potenti e assolutamente inusuali. Basti pensare che le istruzioni di controllo superano in numero e potenza quelle disponibili in Pascal.

### Come funziona

Ultra Basic si attiva come un normale programma Basic. Potete







C64

caricarlo quindi con la usuale istruzione Load e lanciarlo con Run. All'attivazione dell'espansione compare una schermata che, oltre a confermare la presenza del nuovo interprete, mostra la memoria di cui disponete: ben 30719 byte. Praticamente Ultra Basic non ruba neanche un byte alla RAM Basic (inoltre buona parte della RAM in C000 è ancora disponibile) e vi farà anche risparmiare moltissima memoria perché, come vedrete, una sola istruzione di Ultra Basic spesso equivale a parecchie istruzioni del Basic standard.

### Supporto alla programmazione

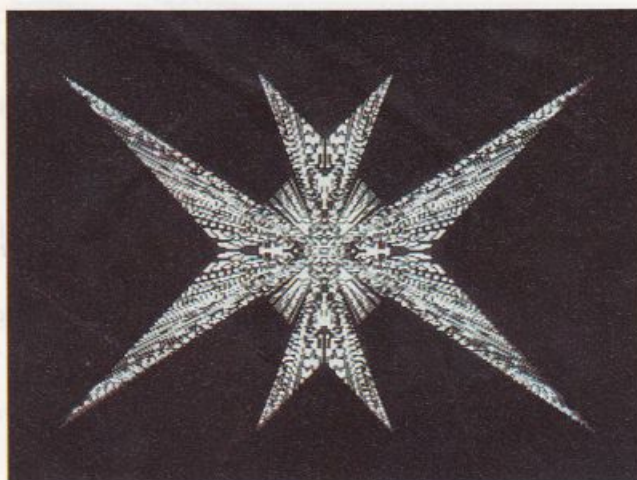
Le istruzioni di questo gruppo sono tra le più importanti, perché permettono di sfruttare pienamente le caratteristiche dell'espansione. Innanzitutto è da segnalare la presenza dell'istruzione Trace, con cui è possibile tracciare l'esecuzione di un programma linea per linea. Altrettanto interessante la presenza delle istruzioni per la ricerca di una stringa all'interno di un programma, per effettuare il dump delle variabili, per attivare il monitor del linguaggio macchina e per delimitare la zona di memoria disponibile per i programmi.

• **Auto:** è un'istruzione utilissima anche se estremamente semplice. Permette di ottenere automaticamente il numero della prossima linea Basic da inserire nel programma. La sintassi di questa istruzione è: auto init, incr. Il parametro init rappresenta il numero di linea da cui deve partire la numerazione automatica, mentre il parametro incr indica l'incremento fra le linee. I due parametri devono essere sempre specificati nell'istruzione, perché non esistono valori di default. Se non vengono indicati entrambi i valori, l'interprete visualizzerà un opportuno messaggio d'errore. Il valore del parametro incr deve essere un intero non negativo (quindi può anche essere 0). Ecco qualche esempio d'uso:

```
auto 10,10
```

(genera numeri di linea a partire da 10 e con incremento 10)

```
auto 100,0
```



(genera sempre il numero di linea 100).

Per disattivare la numerazione automatica dovete premere Return quando compare il numero di linea.

• **Renum:** consente di renumerare le linee del programma in memoria e quindi può essere usata per creare spazio per l'inserimento di nuove linee. La sintassi dell'istruzione è: renum init, incr. Il primo parametro, init, rappresenta il numero di linea da associare alla prima linea del programma (Renum agisce sempre su tutte le linee del programma) mentre incr rappresenta l'incremento fra i numeri di due linee successive. I due parametri possono essere omessi (entrambi contemporaneamente e non solo uno). In questo caso il valore dei due parametri è fissato a 10. Ecco qualche esempio:

```
renum 10,100
```

(renumerare le linee del programma in modo tale che il primo numero di linea sia 10 e che l'incremento fra i numeri di due linee consecutive sia 100).

```
renum
```

(renumerare le linee del programma in modo tale che il primo numero di linea sia 10 e che l'incremento fra i numeri di due linee consecutive sia 10).

• **Delete:** questa istruzione permette di cancellare in un solo colpo più linee di

*La funzione Rec di Ultra Basic può essere utilizzata per creare composizioni di questo tipo.*





**C64**

## ESPANSIONE

programma. La sintassi di questa istruzione è identica a quella dell'istruzione list (con la sola differenza che si deve indicare sempre almeno un parametro). Vediamo qualche esempio:

delete -10

*(cancella tutte le linee fino alla 10 ; la linea 10 è compresa).*

delete 10-190

*(cancella tutte le linee fra la 10 e la 190 ; le linee 10 e 190 sono comprese).*

delete 10-

*(cancella tutte le linee fra la 10 e l'ultima ; la linea 10 è compresa).*

Per cancellare una sola linea dovete usare l'istruzione: delete ln-ln, dove ln è la linea da cancellare (se si digita delete ln l'interprete segnala errore).

- **Old:** annulla l'effetto delle istruzioni new e kill, quindi permette di recuperare l'eventuale programma Basic in memoria perso dopo l'esecuzione di una delle due istruzioni indicate. Questa istruzione non necessita di alcun parametro.

- **Trace:** è l'istruzione più utile e comoda di questo gruppo. Permette infatti di tracciare l'esecuzione del programma Basic in memoria linea per linea. Questo comando va dato in modo diretto e opzionalmente permette di controllare la velocità con cui il programma viene eseguito. La sintassi dell'istruzione è: trace v, dove v rappresenta la velocità di

esecuzione del programma e deve essere un intero compreso fra 0 (velocità maggiore) e 255 (velocità minore). Dando il Run quando è attivo il modo Trace, il programma in memoria viene eseguito alla velocità specificata nell'istruzione trace, e in alto a sinistra sullo schermo viene indicato il numero della linea in esecuzione. Per disattivare il modo Trace dovete usare l'istruzione Strace.

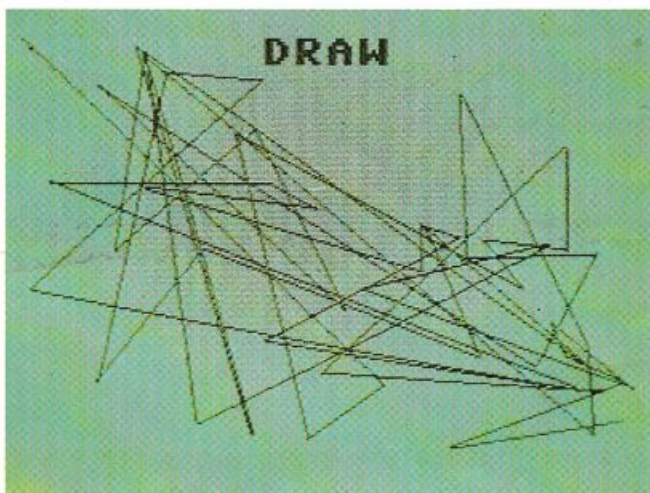
- **Monitor:** è una delle istruzioni più inusuali dell'espansione. Permette di attivare l'eventuale monitor per il linguaggio macchina caricato dopo aver attivato l'espansione. Prima di usare questa istruzione si deve comunicare all'interprete l'indirizzo di inizio del monitor, utilizzando l'istruzione successiva.

- **Set:** comunica all'interprete l'indirizzo di inizio del monitor per il linguaggio macchina. Fissato questo valore (e caricato in memoria il monitor) potete attivare il monitor digitando semplicemente monitor. Per esempio, supponiamo di aver caricato un monitor a partire dalla locazione 20000. Per poter successivamente attivarlo senza l'istruzione sys (che risulta molto scomoda se si dimentica l'indirizzo giusto, oppure se si commette un errore scrivendo l'indirizzo) dovete usare l'istruzione set 20000. Da questo momento per attivare il monitor non dovrete far altro che digitare monitor in modo diretto.

- **Dump:** questa è una delle istruzioni che userete più spesso durante il debugging dei programmi, perché permette di ottenere il dump delle variabili definite nel programma in memoria. Questa istruzione, che non necessita di alcun parametro, visualizza l'elenco completo delle variabili definite nel programma mostrandone il contenuto. La visualizzazione della lista può essere fermata premendo il tasto Shift. Dalla lista sono esclusi gli array.

- **Matrix:** questa istruzione permette di effettuare il dump degli array definiti nel programma Basic in memoria. Utilizzando questa istruzione e la precedente, si può ispezionare in qualunque

La funzione Draw. Per generare questa spezzata ingrovigliata bastano pochissime istruzioni.







istante il contenuto di tutte le variabili del programma. Anche questa istruzione non necessita di alcun parametro.

- **Find:** permette di ricercare una stringa all'interno del programma Basic in memoria. Ciò che si ottiene è l'elenco completo delle linee che contengono la stringa specificata.

La sintassi dell'istruzione è: find stringa, dove stringa rappresenta la stringa da cercare, che deve essere specificata senza usare i doppi apici. Vediamo qualche esempio:

```
find a$
(cerca nel programma Basic in memoria
la stringa a$)
find goto
(cerca nel programma l'istruzione goto)
```

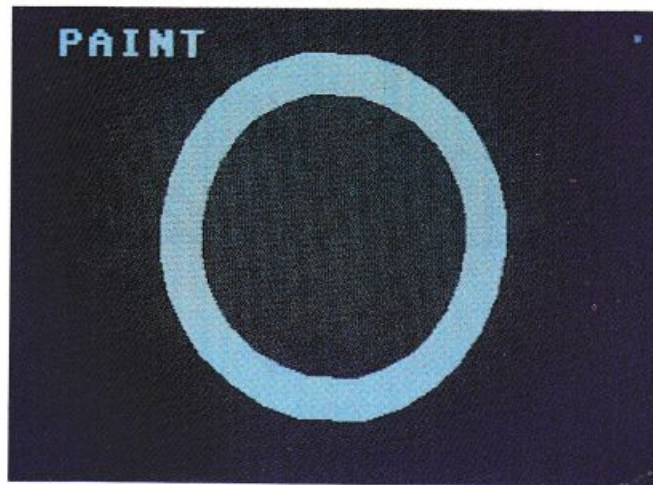
Per specificare la stringa da cercare avete a disposizione anche il metacarattere #, che corrisponde alla stringa di zero o più caratteri qualsiasi. Vediamo qualche esempio:

```
find #a$
(cerca nel programma tutte le variabili
stringa)
find #a
(cerca nel programma tutte le stringhe
che finiscono col carattere a)
find a#
(cerca nel programma tutte le stringhe
che iniziano col carattere a).
```

- **Key:** permette di assegnare una stringa a uno degli 8 tasti funzione a disposizione. La sintassi dell'istruzione è: key n,a\$. Il primo parametro rappresenta il numero del tasto funzione mentre a\$ è la stringa che si vuole associare, che deve essere specificata utilizzando i doppi apici. La lunghezza massima della stringa è di 31 caratteri. Potete inserire anche il carattere Return nella stringa utilizzando il carattere freccia a sinistra (<). Ecco un esempio:

```
key 7,"list<-"
(assegna al tasto funzione F7 la stringa
list + il carattere Return).
```

Premendo il tasto funzione F7 si otter-



rà il list del programma in memoria.

- **Show:** mostra l'elenco delle stringhe associate ai tasti funzione.

- **Off:** annulla tutti gli assegnamenti ai tasti funzione, cioè cancella le stringhe assegnate ai tasti funzione mediante l'istruzione Key.

- **Delay:** permette di fissare la velocità con cui procede il list quando si tiene premuto il tasto Commodore. La sintassi dell'istruzione è: Delay n, dove n è un intero tra 0 (list veloce) e 255 (list lento). Per bloccare temporaneamente il list dovete usare il tasto Shift.

- **Page:** permette di ottenere il list a pagine. La sintassi dell'istruzione è page n, dove n è il numero di linee da visualizzare su ogni pagina e deve essere un intero compreso fra 1 e 25. Visualizzata una pagina, per passare a quella successiva dovete premere la barra di spazio. Per riattivare il list usuale dovete usare l'istruzione Page senza il parametro.

- **Memory:** visualizza la mappa della memoria, cioè la quantità di memoria usata dalle variabili, stringhe e vettori, la quantità di memoria ancora disponibile e quella utilizzata dal programma in memoria.

Daniele Maggio  
(continua)

*La funzione Paint riempie zone del disegno individuate da linee dello stesso colore.*





C64

DIDATTICA

# Le basi per sviluppare in Logo

*Questo corso di Logo è composto da due parti. La prima, che si conclude con questa puntata, vi insegna a sviluppare programmi Logo discretamente complessi, offrendovi solidissime basi per affrontare la seconda parte del corso che inizieremo, al ritorno dalle vacanze estive, sul prossimo numero.*

## Terza puntata

### CAPITOLO SETTIMO

#### Le basi per sviluppare in Logo

Nei capitoli precedenti abbiamo visto come, usando le procedure con parametri, possiamo usare dei nomi al posto di valori nella definizione delle procedure. Un esempio classico è la definizione di Quadrato:

```
PER QUADRATO :LATO :LATO
RIPETI 4 [A :LATO D 90]
FINE
```

Questo è però solo uno dei tipi di variabile che troviamo in LOGO: per distinguerle da altri tipi di variabile, queste variabili vengono chiamate "locali" (in seguito vedremo che cosa significa).

Con l'istruzione Assegna è possibile creare e uti-

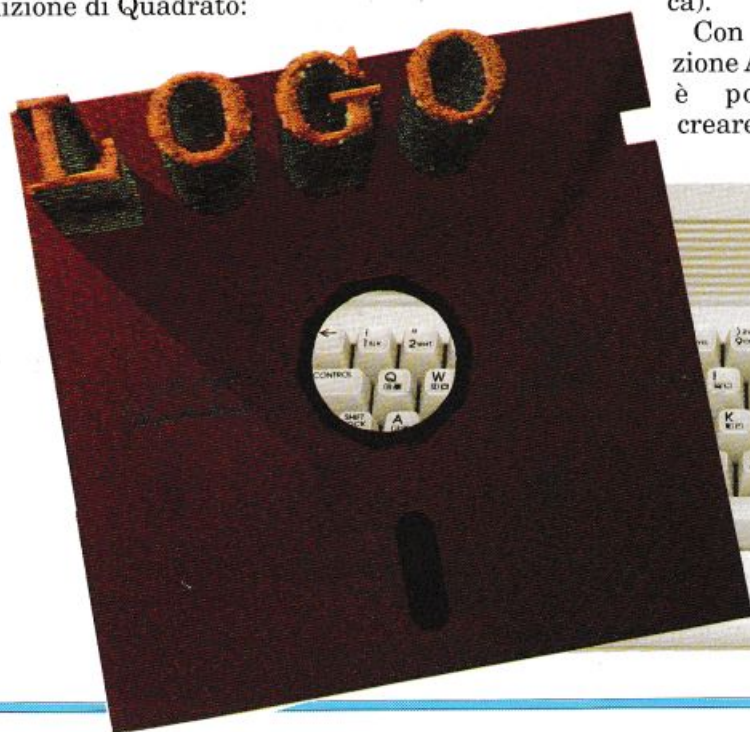
lizzare delle variabili con caratteristiche leggermente diverse da quelle che abbiamo osservato in precedenza.

Provate a digitare questa istruzione in modo diretto:

```
ASSEGNA "AREA 10 * 15
```

Il suo significato è il seguente: crea una variabile di nome Area e assegna il risultato dell'operazione  $10 * 15$ .

```
STAMPA :AREA
```







C64

Avrete notato una stranezza. Con Assegna, per riferirci alla variabile Area, usiamo le virgolette ("), in seguito usiamo i due punti (:). Questa è una caratteristica della sintassi di Logo; quando si usa una variabile per assegnarle un valore è necessario usare le virgolette, quando invece si vuole utilizzare il valore contenuto nella variabile si usano i due punti.

Analizzate il seguente esempio alla luce di quanto appena detto:

```
ASSEGNA "LATOMIN 100
ASSEGNA "LATOMAG 150
ASSEGNA "AREA :LATOMIN *
:LATOMAGG
```

Che risultato darà Stampa :Area? Vediamo l'utilizzo di Assegna all'interno di una procedura:

```
PER DIVIDI :X :Y
ASSEGNA "RISULTATO :X./:Y
STAMPA [IL RISULTATO È]
STAMPA :RISULTATO
FINE
```

Una caratteristica delle variabili create con Assegna è che esse sono delle variabili permanenti: vengono create nel momento in cui viene loro assegnato un valore e rimangono in memoria fino al momento in cui non vengono esplicitamente cancellate.

I parametri che si dichiarano a fianco del nome della procedura si comportano diversamente.

Vediamo un esempio:

```
PER VALORE :NUMERO
STAMPA [IL VALORE DI NUMERO È]
STAMPA :NUMERO
FINE
```

Una volta definita questa procedura, chiamatela così:

```
VALORE 50
LOGO risponderà:
```

IL VALORE DI NUMERO È  
50

Questa conferma che la variabile :Numero contiene 50 all'interno della procedura Valore. Notate bene, solo all'interno della procedura. Difatti, se ora provate a scrivere:

```
STAMPA :NUMERO
```

Logo risponde:

```
MANCA IL NOME
NUMERO
```

Questo significa che la variabile (o nome) Numero non esiste più.

Una caratteristica delle variabili dichiarate a fianco del nome della procedura è infatti di esistere solamente all'interno della procedura dove vengono dichiarate. Per questa loro caratteristica esse vengono chiamate variabili locali.

Vediamo come si comportano le variabili create con Assegna. Scrivete la seguente procedura:

```
PER CUBO :NUMERO
ASSEGNA "RISULTATO :NUMERO *
:NUMERO *
:NUMERO
STAMPA [IL RISULTATO È]
STAMPA :RISULTATO
FINE
```

Chiamatela con:

```
CUBO 3
```

La variabile :Risultato, che viene creata all'interno di Cubo, mantiene il proprio valore anche dopo l'esecuzione della procedura.

Per verificare ciò provate a scrivere:

### Listato 1.

```
PER CERCHIO
RIPETI 180 [A 1 D 2]
FINE

PER SPIRALE :LATO
SE :LATO > 100 STOP
A :LATO
D 72
SPIRALE :LATO + 5
FINE

PER OTTO :LATO
RIPETI 8 [A :LATO D 45]
FINE

PER MENU
STAMPA [MENU GRAFICO, PREMI :]
STAMPA [1 PER DISEGNARE UN CERCHIO]
STAMPA [2 PER DISEGNARE UNA SPIRALE]
STAMPA [3 PER DISEGNARE UN OTTAGONO]
STAMPA [4 PER PULIRE LO SCHERMO]
STAMPA [5 PER TERMINARE]
STAMPA [SCEGLI UN NUMERO TRA 1 E 5]
ASSEGNA "SCELTA LEGGICAR
SE :SCELTA = 1 ALLORA CERCHIO
SE :SCELTA = 2 ALLORA SPIRALE 5
SE :SCELTA = 3 ALLORA OTTO 40
SE :SCELTA = 4 ALLORA DISEGNA
SE :SCELTA = 5 ALLORA STOP
MENU
FINE
```

```
STAMPA :RISULTATO
```

Logo risponde 27. Le variabili di questo tipo vengono chiamate Globali, perché il loro valore rimane permanente nella memoria del computer ed è perciò accessibile all'interno di tutte le procedure.

Vediamo ora qui di seguito un altro esempio dell'uso di Assegna:

```
PER SPIRALE :PASSO
ASSEGNA "ANGOLO 0
RIPETI 1000 [A :PASSO D :ANGOLO
ANGOLO
ASSEGNA "ANGOLO :ANGOLO + 7]
FINE
```

Proviamola con:

```
DISEGNA SPIRALE 10
```

Analizziamone ora il funzionamento passo per passo:

```
ASSEGNA "ANGOLO 0
```





crea la variabile globale Angolo e vi assegna il valore 0;

RIPETI 1000

ripete mille volte;

A :PASSO

sposta la tartaruga in avanti a seconda del valore specificato nella chiamata (Spirale 10);

D :ANGOLO

ruota la tartaruga a destra del valore di Angolo (inizialmente 0);

ASSEGNA "ANGOLO :ANGOLO + 7

assegna alla variabile ANGOLO il valore in essa contenuto più 7 (:Angolo + 7).

La spirale prodotta da questa procedura è molto interessante: provate a modificare sia il passo

della tartaruga (Spirale 15, eccetera), sia l'incremento dell'angolo (:Angolo + 9) e studiatene l'effetto.

Finora abbiamo usato le variabili solo per memorizzare dei numeri. Esse però possono contenere anche altri oggetti come Parole e liste. Il modo di procedere è esattamente lo stesso di quello usato per i numeri:

```
ASSEGNA "NOME "MARIO
STAMPA :NOME
MARIO
```

```
ASSEGNA "AMICI [GINO GIULIO GIANNI]
STAMPA :AMICI
GINO GIULIO GIANNI
```

Fate attenzione a non fare confusione con le virgolette.

Nel primo esempio le virgolette davanti al nome indicano la variabile, quelle davanti al nome Mario dichiarano a Logo che

quella è una parola.

La capacità di memorizzare tutti questi tipi di informazioni è una caratteristica molto interessante di Logo, che apre spazio alle applicazioni più svariate.

Nell'ultimo capitolo vedremo come costruire un piccolo archivio bibliografico.

Ecco la procedura che potrebbe rappresentare la base di un mini-archivio.

```
PER NUOVAMICO :NOME
SE :NOME = " ALLORA STOP
ASSEGNA "AMICI INULT :NOME :AMICI
FINE
```

```
PER LISTAAMICI
STAMPA :AMICI
FINE
```

La procedura NuovoAmico inserisce il nome di un amico in coda alla lista di amici precedentemente creata con l'istruzione:

### Listato 2.

```
PER OROLOGIO
NT ;nasconde la tartaruga
QUADRANTE
```

```
ASSEGNA "MINUTI 0 ;regola l'ora
TICTAC
FINE
```

```
PER QUADRANTE
RIPETI 12 [SEGNO D 360/12]
FINE
```

```
PER SEGNO
SU A 50 GIU A 10 SU I 60 GIU
FINE
```

```
PER SFERA.MINUTI
A 40 I 40
FINE
```

```
PER DISEGNA MINUTI
TANA
D :MINUTI * 6 ;360/60 fa 6 gradi
ASCOL 1
SFERA.MINUTI
FINE
```

```
PER CANCELLA.MINUTI
TANA
D :MINUTI * 6
ASCOL -1 ;cancella
SFERA.MINUTI
```

FINE

```
PER SFERA.ORE
A 20 I 20
FINE
```

```
PER DISEGNA.ORE
TANA
D :MINUTI / 2 ;1 grado ogni 2 minuti
ASCOL 1
SFERA.ORE
FINE
```

```
PER CANCELLA.ORE
TANA
D :MINUTI / 2
ASCOL -1
SFERA.ORE
FINE
```

```
PER MINUTO
RIPETI 51000 [] ;regola la pausa
FINE
```

```
PER TICTAC
CANCELLA.ORE
CANCELLA.MINUTI
AS "MINUTI :MINUTI + 1 ;ricorda il tempoDISEGNA.MINUTI
DISEGNA.ORE
MINUTO ;aspetta 1 minuto
TICTAC
```





**ASSEGNA "AMICI [GINO GIULIO GIANNI]**

Per questa operazione viene usata l'istruzione Inult, (che significa INserisci all'ULTimo posto). La lista di partenza è formata dai nomi [Gino Giulio Gianni]. Chiamando la procedura in questo modo:

**NUOVOAMICO "ANTONIO**

La lista di amici sarà la seguente:

[GINO GIULIO GIANNI ANTONIO]

Per verificarlo basta usare ListaAmici. Dato che le variabili create con Assegna sono permanenti, occupano stabilmente una parte della memoria nel computer. Quando il loro utilizzo è terminato, è perciò conveniente eliminarle.

Per questo motivo Logo mette a disposizione alcune istruzioni per vedere le variabili che esso conosce in un determinato istante e altre per cancellarle.

L'istruzione FaiVedere seguita dal comando Nomi, mostra una lista di tutte le variabili finora utilizzate e il loro contenuto:

**FAIVEDERE NOMI**

o, abbreviando:

**FV NOMI**

L'istruzione CanceNomi, seguita dal nome di una variabile, la cancella dalla memoria del computer.

Fate attenzione perché questa operazione è definitiva, il contenuto di una variabile cancellata non può essere recuperato. Esempio:

**ASSEGNA "COLORE "ROSSO  
FV NOMI**

**Esercizi - Capitolo settimo**

1) Come approfondimento scoprite la funzione della primitiva Locale (Per esempio LOCALE "VAR1 AS "VAR1 2500).

*N.B. Locale viene usato solo all'interno di procedure.*

2) Definite una serie di variabili che vi permettono di cambiare il colore della tartaruga utilizzando il nome del colore invece di un numero (Per esempio: ASCOL :ROSSO).

(appare la variabile COLORE nella lista)  
CANCENOMI "ROSSO  
FV NOMI  
(la variabile ROSSO non compare più).

**CAPITOLO OTTAVO**

**Comunicare con l'esterno**

I programmi di Logo che abbiamo scritto finora avevano alcune informazioni in Input che venivano specificate nella chiamata della procedura. Per esempio, in:

**QUADRATO 50**

Il numero 50 è il dato in input che viene inserito nella variabile :Lato e in seguito viene utilizzato con l'istruzione A :Lato.

Esistono però altri modi di inserire informazioni nelle variabili. A volte infatti è comodo che il computer ci faccia esplicitamente delle domande, chiedendoci le informazioni che gli servono. Questo metodo di programmazione semplifica l'uso del computer da parte di persone inesperte (tutti sono in grado di rispondere a semplici domande che appaiono sullo schermo del computer). Vediamo un esempio:

**PER NUOVOQUA  
STAMPA [INSERISCI LA LUNGHEZZA DEL LATO]  
ASSEGNA "LATO PRIMO LEG-**

GIRIG  
RIPETI 4 [A :LATO D 90]  
FINE

Una volta definita la procedura chiamatela con:

**NUOVOQUA**

Logo risponde:

INSERISCI LA LUNGHEZZA DEL LATO

Rispondete:

50 <RETURN>

A questo punto la tartaruga dovrebbe disegnare un quadrato di lato 50. Questa procedura è in parte molto simile a quella di un semplice quadrato che già conosciamo. La parte nuova è quella che definisce la lunghezza del lato del quadrato da disegnare (contenuta nella variabile LATO). Questo compito è svolto dall'istruzione:

**ASSEGNA "LATO PRIMO LEGGIRIG**

Possiamo scomporre questa istruzione in tre parti:

- **LEGGIRIG:** con questa istruzione, Logo si ferma e attende che noi inseriamo un'informazione dalla tastiera. Nel nostro esempio abbiamo inserito





**C64**

## Didattica

50. Il numero 50 viene perciò letto dalla tastiera e inserito in una lista, viene cioè racchiuso tra parentesi quadre. Esso si presenta così [50].

- **PRIMO**: questa istruzione, come abbiamo visto precedentemente, serve per isolare il primo elemento di una lista, nel nostro caso gli viene passata la lista [50] che contiene un solo elemento. Il risultato è il numero 50.

- **ASSEGNA:LATO**: questa sequenza inserisce il numero 50 (ri-cavato dalle due istruzioni precedenti) nella variabile Lato.

In pratica nel nostro esempio la sequenza è equivalente a scrivere **ASSEGNA:LATO 50**, solo che il numero 50 viene inserito soltanto al momento della chiamata dalla procedura. Provate a chiamare più volte la procedura **NuovoQua** e a rispondere con numeri diversi:

```
NUOVOQUA
INSERISCI LA LUNGHEZZA
DEL LATO
70 <RETURN>
(disegna un quadrato di lato 70)
```

Verificate che il numero inserito venga memorizzato nella variabile Lato. Per fare questo scrivete:

### FV NOMI

Tra le variabili apparirà Lato con associato l'ultimo numero che avete inserito. Ciò che abbiamo imparato a fare con l'uso di **LeggiRig** è la costruzione di procedure interattive; vale a dire procedure che procedono per domande e risposte. Vediamo un altro esempio:

```
PER INCONTRO
STAMPA [CIAO COME TI
CHIAMI?]
ASSEGNA "NOME PRIMO
LEGGIRIG
(STAMPA [PIACERE DI CO-
NOSCERTI,]:NOME)
STAMPA [IO MI CHIAMO C64]
FINE
```

Una volta definita provatela così:

```
INCONTRO
CIAO COME TI CHIAMI?
UGO <RETURN> (questo lo do-
vete scrivere voi)
PIACERE DI CONOSCERTI
```

## UGO IO MI CHIAMO C64

Le prospettive che offrono questo tipo di istruzioni sono senz'altro interessanti; provate a pensare anche voi ad alcune semplici applicazioni. Nell'ultimo capitolo verranno usate per la costruzione del progetto archivio.

Il **listato 1** riporta un altro esempio di costruzione di procedure interattive. Lo scopo di questa serie di procedure è di automatizzare il disegno di alcune figure. Una volta definite tutte le procedure, chiamate **Menù**. A questo punto potete scegliere tra 5 operazioni da eseguire. La scelta viene effettuata mediante la semplice pressione di un tasto (i numeri da 1 a 5). L'istruzione che ha permesso di fare questo è **LeggiCar**. Essa si comporta in modo simile a **LeggiRig** ma, come suggerisce il nome, invece di leggere un'intera riga dalla tastiera legge un solo carattere per volta.

Dato che legge un solo carattere alla volta non è necessario premere **Return** per terminare l'inserimento. Basta premere il numero corrispondente alla scelta.

## CAPITOLO NONO

### Un progetto

Sebbene nel corso dei precedenti capitoli abbiate elaborato delle procedure anche complesse, non avevate a disposizione la conoscenza di tutte le primitive e di tutte le regole per elaborare un progetto organico, che non fosse fine a se stesso, ma potesse venir usato per successive esplorazioni nel mondo del Logo.

In questo e nel prossimo capitolo, affronterete la progettazione di due super-procedure che vi consentiranno interessanti impieghi di questo linguaggio, oltre a costituire una sintesi di quanto espresso.

### Listato 3.

```
PER NUOVOLIBRO
AS "LIBRO []
CHIEDI [NOME E COGNOME DELL'AUTORE ?]
AS "LIBRO INULT :RISPOSTA :LIBRO
CHIEDI [TITOLO DEL LIBRO ?]
AS "LIBRO INULT :RISPOSTA :LIBRO
CHIEDI [CASA EDITRICE ?]
AS "LIBRO INULT :RISPOSTA :LIBRO
CHIEDE [ANNO DI PUBBLICAZIONE ?]
AS "LIBRO INULT :RISPOSTA :LIBRO
CHIEDI [ANNOTAZIONI ?]
AS "LIBRO INULT :RISPOSTA :LIBRO
AS "ARCHIVIO INULT :LIBRO :ARCHIVIO
ST [INSERIMENTO TERMINATO !]
FINE PER CHIEDI :MESSAGGIO
ST :MESSAGGIO
AS "RISPOSTA LEGGIRIG
SE :RISPOSTA = [] ALLORA CHIEDI :MESSAGGIO
SE :RISPOSTA = [STOP] ALLORA ST [INSERIMENTO INTERROTTO]
PUNTOCAPO
FINE
```





**Listato 4.**

```

PER LISTALIBRI
LISTA2 :ARCHIVI
FINE

PER LISTA2 :LIBRI
SE :LIBRI = [] STOP
FVLIBRO PRIMO :LIBRI
LISTA2 MENPRI :LIBRI
FINE
PER FVLIBRO :LIBRO
(ST [AUTORE] -] ELEMENTO :AUTORE :LIBRO)
(ST [TITOLO -] ELEMENTO :TITOLO :LIBRO)
(ST [CASAED -] ELEMENTO :CASAED :LIBRO)
(ST [ANNO -] ELEMENTO :ANNOPB :LIBRO)
(ST [NOTE -] ELEMENTO :NOTE :LIBRO)
FINE
    
```

Lo scopo del primo progetto è costruire un orologio analogico, uno di quelli con le lancette, e farlo funzionare. Se ci pensate un attimo vi accorgete che il problema può essere scomposto in tre parti: il quadrante, le lancette e il funzionamento; quindi potete affrontarle e risolverle separatamente.

Nel quadrante la parte veramente importante sono i segni delle ore, normalmente rappresentati da punti o da brevi segmenti.

Una possibile soluzione per disegnare il quadrante consiste nel costruire una procedura che partendo dal centro disegni un segmento ad una certa distanza. Provate da soli:

- nome consigliato: **SEGNO**;
- primitive consigliate: **TANA, SU, GIU, A, I**.

Il secondo passo consiste nel ripetere questa procedura per 12 volte lungo un'immaginaria circonferenza. Non è difficile: in pratica è come dire alla tartaruga di spostarsi ogni volta di un dodicesimo di circonferenza (cioè di 360/12 gradi) prima di disegnare il segmento successivo. Provate da soli:

- nome consigliato: **QUADRANTE**

La procedura Quadrante comporta che la tartaruga parta dal centro e si trovi, alla fine, sempre nel centro.

Per le lancette sembra che si possa fare altrettanto. Provate da soli:

- nomi consigliati:  
**SFERA.MINU-  
TI SFERA.ORE**

Rimane il problema dello spostamento delle lancette, cioè della misura del tempo vera e propria: è una procedura complessa, formata dal disegno delle lancette e dal conteggio dell'intervallo di un minuto. Per ora e per comodità consideriamola già risolta e vediamo come dovrebbe comportarsi. Chiamiamola TicTac.

Nel modello funzionante, TicTac dovrebbe venir eseguita per tutto il tempo in cui l'orologio è in funzione. Per esempio, con un:

```
RIPETI 1000 [TICTAC]
```

In questo modo l'orologio si fermerà dopo 1000 minuti di funzionamento: per avere una durata maggiore basterà aumentare l'argomento di Ripeti. E adesso è ora di definire TicTac; per descrivere il movimento delle ore e dei minuti conviene usare una variabile (Minuti). Ecco cosa dovrebbe fare TicTac:

- cancellare la sfera dei minuti;
- cancellare la sfera delle ore;
- aggiornare la variabile minuti;
- disegnare la sfera dei minuti nella nuova posizione;
- disegnare la sfera delle ore.

La nuova posizione per la sfera dei minuti si può ottenere calco-

lando lo spostamento angolare della tartaruga con 360/60; mentre per le ore il calcolo è 360/12. Provate.

Se va tutto bene l'orologio funziona ma è troppo veloce: bisogna trovare un modo per far sì che il computer, prima di disegnare la lancetta dei minuti in una nuova posizione, lasci passare un minuto. Tenete presente che, per la velocità che sia, il computer impiega comunque del tempo nello svolgere il suo lavoro, per esempio per contare o per eseguire una procedura. Perché, allora, tra un disegno di lancette e un altro, non fargli fare qualcosa che duri un minuto e che nello stesso tempo non interferisca con il disegno? Un'attività che sicuramente non interferisce con niente è quella di contare. Provate questa:

```

PER ATTESA :TEMPO
RIPETI :TEMPO []
FINE
    
```

Chiamate Attesa 6000 e osservate che non succede niente eccetto il fatto che il cursore appare solo dopo un certo tempo. Provate con un valore più alto. Il cursore impiega sempre di più a riapparire: il computer, mentre ripete da 1 a 1000 l'istruzione vuota [], impiega del tempo. Quanto, esattamente, potrete trovarlo sperimentalmente. Pur con delle piccole differenze tra un computer e l'altro, il C-64 impiega 1 minuto a contare da 1 a 51000.

Ritornando allora all'orologio si tratta di inserire dopo ogni disegno di lancette la procedura Ripeti....[].

A questo punto potete provare da soli e poi confrontare la vostra soluzione con quella che abbiamo trovato noi (**listato 2**). Che ve ne pare? Naturalmente si può migliorare, per esempio con questi interventi:





**C64**



- modificate il quadrante da rotondo a esagonale;
- aggiungete il valore numerico ai segni delle ore;
- aggiungete un quadrante con lancetta per i secondi;
- fate in modo di poter assegnare una certa ora di partenza;
- aggiungete un quadrante a lettura digitale per i secondi;
- trasformatelo in cronografo;
- fatelo con solo 6 ore, oppure con 13 ore oppure....;
- fatelo che segni il tempo a ritroso;
- e chi più ne ha, più ne metta.

## CAPITOLO DECIMO

### Un altro progetto

In quest'ultimo capitolo affronteremo un progetto un po' ambizioso: costruire un programma per gestire un archivio bibliografico.

In effetti costruire un vero archivio elettronico sarebbe un'impresa troppo impegnativa, ciò

che faremo noi è solo un piccolo esempio. Esso ci aiuterà comunque a capire come sono organizzati questi tipi di programma e a migliorare la nostra abilità di programmazione in Logo.

Nella costruzione di un archivio bisogna pianificare alcune cose prima di affrontare la programmazione vera e propria. In particolare è necessario determinare che tipo di informazione vogliamo memorizzare nell'archivio che andiamo creando. Nel nostro esempio vogliamo memorizzare alcune informazioni riguardanti dei libri. Dobbiamo decidere quali delle molte informazioni che riguardano i libri ci interessa memorizzare.

Per questo esempio teniamo solo le seguenti informazioni:

- autore del libro;
- titolo del libro;
- casa editrice;
- anno di pubblicazione;
- note.

Probabilmente alcuni di voi che hanno interessi specifici vorrebbero inserire altri tipi di informazioni: non preoccupatevi, una volta capito il meccanismo sarete in grado di modificare la struttura dell'archivio a vostro piacimento.

Dopo aver definito le informazioni da memorizzare, dobbiamo dare loro un nome e un numero di riferimento (o posizione):

AUTORE	1
TITOLO	2
CASAED	3
ANNOPB	4
NOTE	5

Per ogni libro che verrà memorizzato dovranno essere inserite queste cinque informazioni; esse verranno inserite in una lista.

[[S.PAPERT] [MINDSTORMS]  
[EMME] [1984] [MOLTO INTERESSANTE]]

Questa è la forma con cui ciascun libro apparirà in memoria. Notate che la sequenza delle informazioni segue l'ordine specificato (Autore 1...., Note 5). È infatti determinante che le informazioni vengano inserite al posto giusto.

Le informazioni di ciascun libro verranno poi inserite in un'unica grande lista (l'archivio sarà formato da una lista di liste):

[[libro1] [libro2]...[libro100]]

Naturalmente ogni lista riguardante un libro avrà la struttura specifica sopra (5 elementi). Non fatevi spaventare da tutte queste premesse teoriche, che sono comunque necessarie: la realizzazione pratica è molto più divertente.

Definiamo la prima procedura che ci serve per creare l'archivio:





PER CREARCHIVIO  
AS "ARCHIVIO []  
CREANOMI  
ST [HO CREATO LA STRUTTU-  
RA DELL'ARCHIVIO]  
FINE

PER CREANOMI  
AS "AI UTORE 1  
AS "TITOLO 2  
AS "CASAED 3  
AS "ANNOPB 4  
AS "NOTE 5  
FINE

Con queste due prime procedure creiamo l'associazione tra il nome dell'informazione e la sua posizione, inserendo in una variabile che porta il nome dell'informazione il numero che rappresenta la sua posizione (CreaNomi).

(N.B. AS è l'abbreviazione di Assegna).

Nella procedura CreArchivio viene inoltre creata la lista che conterrà l'archivio, che si chiama appunto Archivio. Essa viene creata vuota (il simbolo [] significa lista vuota).

Definiamo ora la procedura più importante, quella che ci serve per inserire un libro in archi-

vio (vedi **listato 3**).

Le informazioni riguardanti il libro vengono in un primo momento inserite nella lista contenuta nella variabile Libro. L'istruzione:

AS "LIBRO []

prepara la variabile inserendovi una lista vuota. La variabile viene in seguito riempita con le informazioni previste.

La procedura NuovoLibro usa la sottoprocedura Chiedi che serve semplicemente per visualizzare un messaggio e ricevere la risposta. L'informazione inserita viene memorizzata nella variabile Risposta.

Nella procedura Chiedi viene utilizzata l'istruzione:

PUNTOACAPO

essa significa: ferma l'esecuzione di tutte le procedure attive. In questo caso viene usata per permettere di interrompere l'inserimento (provate a rispondere Stop a una domanda).

Dopo aver eseguito la procedura Chiedi, l'informazione (che è contenuta in Risposta) viene

messa nella variabile Libro con la seguente istruzione:

AS "LIBRO INULT :RISPOSTA :LIBRO

In questo modo, alla fine della sequenza di domande e risposte, la variabile Libro conterrà le 5 informazioni previste.

Se avete definito le procedure viste finora, provate a inserire alcune informazioni. Per prima cosa chiamate la procedura Archivio:

CREARCHIVIO <RETURN>

Questa procedura va eseguita una volta sola, al momento della creazione dell'archivio; essa cancella infatti tutte le informazioni contenute nell'archivio. State dunque ben attenti a non richiamarla una volta inserite le informazioni nell'archivio.

Ora è tutto pronto per iniziare l'inserimento.

NUOVOLIBRO <RETURN>

Rispondete alle domande che vi vengono poste. Ricordatevi di premere Return dopo ogni informazione.

Una volta inserite le informazioni riguardanti alcuni libri avrete voglia di verificare che siano state memorizzate correttamente; per verificare ciò provate a scrivere:

STAMPA :ARCHIVIO

Vi apparirà una lunga lista contenente le informazioni che avete appena inserito. Notate come le informazioni sono tutte disposte secondo l'ordine prestabilito e cioè : Autore, Titolo, e così via.

### Esercizi - Capitolo decimo

1) *Definite una procedura che, tramite una serie di domande (usate LeggiRig), chieda: Lato1, Lato2 e Altezza di un parallelepipedo e ne calcoli l'area di base e volume. (Se la vostra fantasia vi suggerisce problemi stimolanti sull'uso di questa tecnica, non esitate a sperimentare nuove soluzioni).*

2) *Provate a progettare altre procedure interattive sul modello di Incontro. Che cosa ne pensate di un incontro con il medico o di un'agenzia matrimoniale?*

3) *Estendete la procedura Menù in modo da comprendere le procedure che disegnano le vostre figure preferite. Lo schema è infatti abbastanza semplice. Basta aggiungere le nuove procedure alla lista di quelle esistenti e allargare in corrispondenza le possibilità di scelta.*





Questo modo di vedere la lista dei libri non è molto soddisfacente. La lista appare disordinata e difficile da leggere. Costruiamo allora una procedura che ci mostri l'archivio in modo più ordinato (vedi **listato 4**).

La procedura da chiamare in questo caso è ListaLibri, che è la procedura principale. La sua funzione è di richiamare LISTA2, che usando una tecnica ricorsiva passa uno alla volta alla sottoprocedura FVLibro i libri contenuti in archivio.

È compito della procedura FVLibro stampare ordinatamente le informazioni inerenti ciascun libro. Per provare queste procedure chiamate:

```
LISTALIBRI <RETURN>
```

Oltre a vedere il contenuto dell'archivio, a volte è molto utile poter interrogare l'archivio per cercare delle informazioni.

Questa è la funzione svolta dalle ultime due procedure che definiamo in questo progetto.

```
PER CERCA :DOVE :INFORMAZIONI  
CERCA2 :ARCHIVIO  
FINE
```

```
PER CERCA2 :LOBRI  
SE :LIBRI = [] STOP  
SE APPARTIENE? :INFORMAZIONI (ELEMENTO :DOVE  
PRIMO :LIBRI)  
ALLORA FVLIBRO PRIMO :LIBRI  
CERCA2 MANPRI :LIBRI  
FINE
```

Questa procedura funziona in maniera leggermente più complessa delle altre.

Il suo compito è quello di trovare tutte le informazioni riguardanti un libro a partire da una sola specifica. Per fare ciò essa ha bisogno di avere in ingresso la posizione dove cercare (Autore o Ti-

tolo o...) e l'informazione che si cerca. Per esempio:

```
CERCA :AUTORE "S.PAPERT
```

Vediamo comunque qual'è la funzione delle istruzioni principali.

L'istruzione più importante è:

```
SE APPARTIENE? :INFORMAZIONE (ELEMENTO :DOVE  
PRIMO :LIBRI)  
ALLORA LISTALIBRO PRIMO :LIBRI
```

Essa può essere scomposta come segue:

```
(ELEMENTO :DOVE PRIMO :LIBRI)
```

Questa sequenza estrae dalla variabile Libri (che contiene temporaneamente l'archivio) il primo elemento (Primo :Libri); vale a dire le informazioni relative ad un libro. Per esempio:

```
[[S.PAPERT] [MINDSTORMS]  
[EMME] [1984] [MOLTO INTERESSANTE]
```

Dopo l'esecuzione di Primo :Libri l'istruzione potrebbe apparire così:

```
(ELEMENTO :DOVE [[S.PAPERT] [MINDSTORMS] [EMME] [1984] [MOLTO INTERESSANTE]
```

L'istruzione Elemento :Dove estrae l'informazione desiderata.

Nel nostro caso l'Autore (il primo elemento). Il risultato sarà perciò:

```
[S.PAPERT]
```

A questo punto viene eseguito:

```
APPARTIENE? :INFORMAZIONE [S.PAPERT]
```

L'istruzione Appartiene verifica se il primo argomento (in questo caso il contenuto di Informazione) è contenuto nel secondo argomento. Se questa condizione è vera il libro è quello che stiamo cercando e perciò viene visualizzato con:

```
ALLORA FVLIBRO PRIMO :LIBRI
```

Avrete notato che anche in questo caso viene usata una tecnica ricorsiva per analizzare uno per uno tutti i libri dell'archivio.

Divertitevi a inserire alcuni libri in archivio e a cercare le informazioni secondo diversi criteri. Potete cercare per anno con:

```
CERCA :ANNOPB "1984
```

oppure per titolo con:

```
CERCA :TITOLO "MINDSTORM
```

Questo progetto era abbastanza ambizioso, però è stato di sicuro stimolante a creare un proprio archivio elettronico. Data la difficoltà delle tecniche utilizzate, non dovete stupirvi se non avete capito tutto; con l'aumentare della vostra esperienza potrete ritornare su queste procedure e sviscerarne tutti i segreti più nascosti.

Prima di spegnere il computer vi consigliamo di salvare su un dischetto il vostro lavoro con l'istruzione:

```
CONSERVA "ARCHIVIO
```

In questo modo vengono salvate le procedure che avete inserito e anche le informazioni che avete in archivio.

La prossima volta che lavorate con Logo potrete richiamare il tutto con:

```
RECUPERA "ARCHIVIO
```

**Fernando Zanini**

(continua)





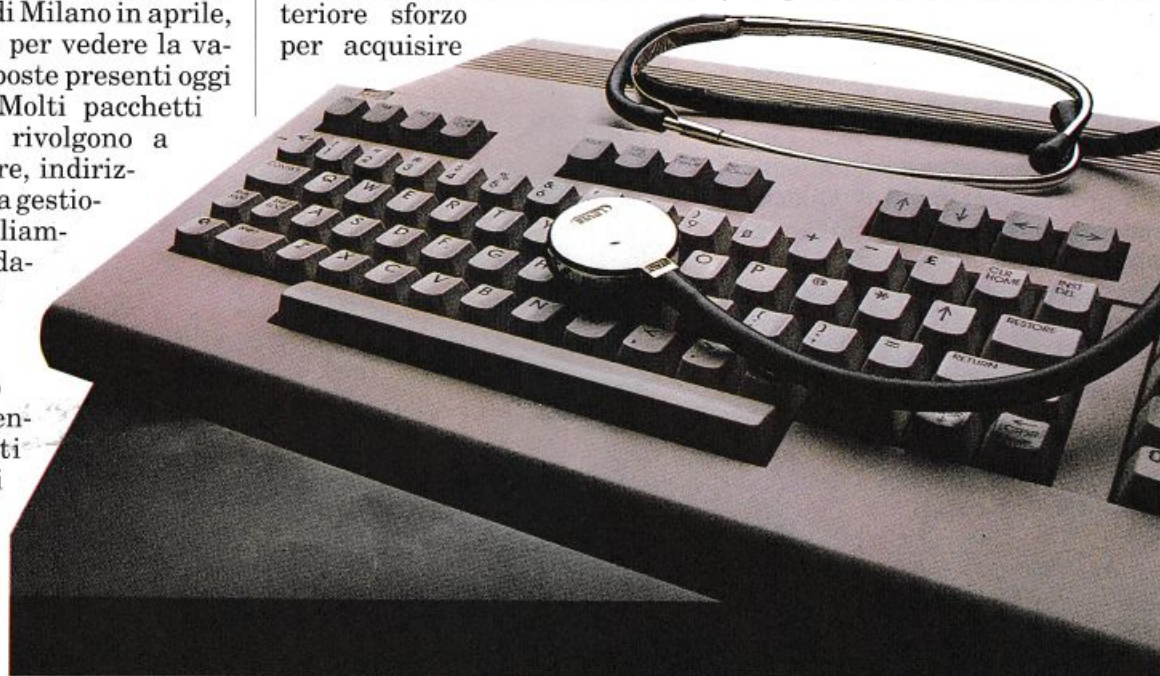
# Una cartella clinica elettronica

*Ogni medico di base, per svolgere seriamente il proprio lavoro, deve costruirsi un archivio contenente la storia sanitaria dei propri assistiti. Queste operazioni possono essere svolte con più efficacia da un programma ottimizzato per questo scopo.*

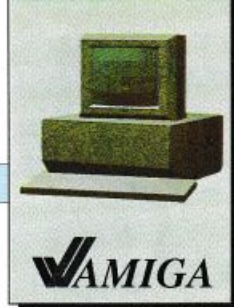
L'esigenza principale di ogni produttore di software è quella di scoprire nuove esigenze del mercato e cercare di soddisfarle. Un settore su cui da tempo si stanno concentrando molti sforzi è certamente l'ambito sanitario, sia perché si presta bene ad alcuni tipi di soluzione informatica, sia per le enormi prospettive di mercato che si apriranno per chi riuscirà a imporsi. Una rapida occhiata alla rassegna Smart Hospital, allestita all'interno della Grande Fiera di Milano in aprile, era sufficiente per vedere la varietà delle proposte presenti oggi sul mercato. Molti pacchetti applicativi si rivolgono a grosse strutture, indirizzandosi verso la gestione di USL, poliambulatori, ospedali, laboratori d'analisi, farmacie, eccetera; altri però sono decisamente indirizzati verso piccoli utenti, come gli specialisti o i medici di base. I sistemi prevalenti anche

in questi ambiti sono gli Ms-Dos compatibili, che però, a fronte della loro capillare diffusione, hanno sempre difettato di un'interfaccia con l'utente semplice e intuitiva, come l'interfaccia grafica di Amiga. Questa attenzione all'utente e alle sue conoscenze informatiche è un elemento indispensabile per ottenere successo in un settore dove il medico fa già molta fatica a mantenersi aggiornato sulle principali patologie, al punto da proibirgli ogni ulteriore sforzo per acquisire

quel minimo di dimestichezza informatica necessaria per sfruttare appieno ogni programma che decida di utilizzare. Inoltre, dopo i primi facili entusiasmi soprattutto sulla diagnosi assistita dal computer, ci si è resi conto che questo doveva svolgere un ruolo di supporto e non di sostituzione dell'attività del medico, caricandosi di tutte quelle operazioni che per la loro ripetitività o per la loro vastità sono più difficilmente gestibili da un essere umano.







AMIGA È...

AMIGA

### L'archivio del medico

Amiga-Medico, sviluppato da Giocologico di Treviso, si propone come strumento capace di aiutare il medico di base a conservare le informazioni relative ai propri pazienti.

Le operazioni di normale gestione sono eseguibili tramite mouse e sono autoesplicanti, in quanto rappresentate in forma grafica e simbolica. La versione completa contiene, oltre agli elementi presenti nel dimostrativo analizzato, programmi di supporto di statistica e di manutenzione.

L'attuale versione è stata scritta in linguaggio C, progettata e realizzata tramite il costante confronto con medici di base; necessita di un Commodore Amiga con almeno 512 Kb di RAM e di due drive o di un hard disk. Il dimostrativo è liberamente distribuibile.

Scopo del programma è di rimpiazzare l'archivio cartaceo che ogni medico possiede e in cui registra tutti gli eventi connessi alla sua relazione con il paziente. All'apertura del programma pertanto si presenta la barra dei menù, che permette con la funzione Archivio di accedere ai vari cassette, contrassegnati ognuno da una diversa lettera dell'alfabeto (ci sono anche due cassette speciali che possono essere impiegate a proprio piacimento). Questa operazione corrisponde all'aper-

tura di un cassetto, ed è indispensabile eseguirla perché indica al programma su quale cassetto si applicheranno tutte le sue funzioni. Anche se non si dispone di un hard disk, ma si è costretti a utilizzare numerosi floppy, non ci saranno problemi nel reperire un certo paziente perché il programma si incaricherà di indicare il dischetto desiderato.

Dopo aver aggiornato la data perché sia proposta automaticamente in tutte le operazioni che verranno eseguite nella giornata, si passa alla selezione della cartella clinica del paziente appena entrato. L'operazione si compie molto velocemente con il mouse e permette di non dedicare troppa attenzione al calcolatore sottraendola al paziente. Quest'ultimo infatti deve sentirsi accolto come persona dal medico, e non certo considerato come una macchina da aggiustare, il cui catalogo dei pezzi di ricambio viene sfogliato col computer.

La cartella del paziente, una volta aperta, presenta nella metà

superiore tutti i suoi dati anagrafici, compreso il suo codice e il numero dell'esenzione ticket, e le date di inizio ed eventuale termine del rapporto con il malato (figura 1). Poiché non sempre le informazioni importanti possono essere codificate nei ristretti e rigidi spazi di un modello standard, potete aggiungerle sotto forma di promemoria richiamabile con un clic.

Nella metà inferiore della cartella sono rappresentati in forma grafica gli eventi sanitari della vita del paziente; viene visualizzato un periodo di 500 giorni, ma vi potete facilmente spostare con il mouse in qualunque periodo della vita. Gli eventi sono rappresentati in base al tipo di evento o al tipo di patologia: con una semplice pressione del mouse potete passare da una all'altra.

### Gli eventi sanitari

L'elemento chiave di Amiga-Medico è l'evento: in esso sono conservate tutte le informazioni

Figura 1. La cartella clinica di un paziente.





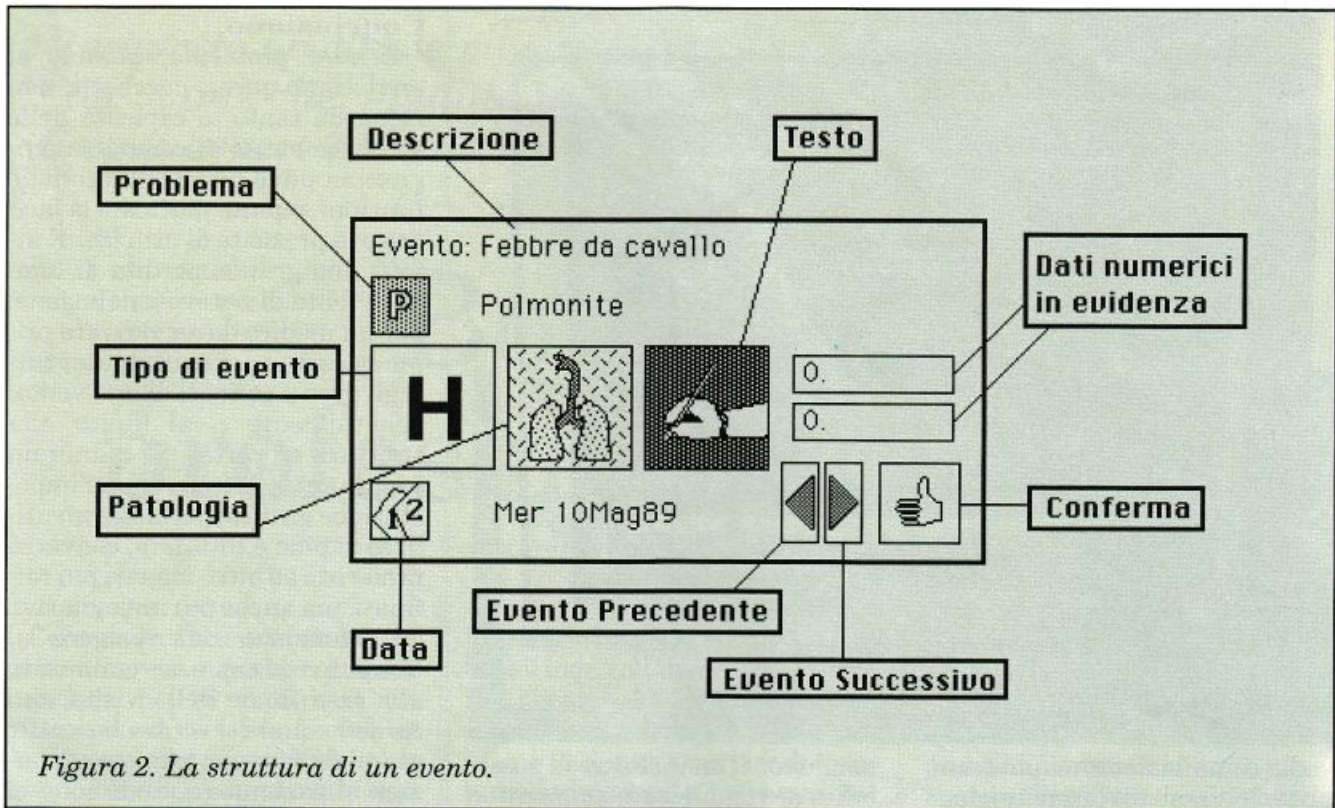


Figura 2. La struttura di un evento.

riguardanti la salute del paziente. L'evento deve essere facilmente definibile, altrimenti la quantità di tempo necessario per quest'operazione scoraggerà chiunque dall'eseguirlo, rendendo inutile il programma. All'interno di ogni box descrittivo di un evento le informazioni sono raffigurate in forma grafica, come piccole immagini o bollini, allo scopo di aiutare il medico a orientarsi immediatamente. Vi è la descrizione di un problema, scelto fra quelli appropriatamente definiti per quel paziente, il tipo di evento, scelto fra 16 diversi (telefonata, prescrizione di farmaci, ricovero, eccetera), il tipo di patologia o di apparato interessato, anch'esso scelto tra 16 diversi, la possibilità di evidenziare alcuni valori (pressione, analisi del sangue, delle urine, eccetera) e infine la possibilità di scrivere una piccola nota con osservazioni particolari (figura 2).

Fra i bollini presenti ve ne sono

due che rappresentano funzioni particolarmente interessanti: uno serve a richiamare il medico a prendere visione delle informazioni in esso contenute prima di procedere nella visita o nella prescrizione, perché contiene informazioni vitali al riguardo (ipersensibilità individuale a farmaci, recenti terapie farmacologiche che invaliderebbero i risultati di analisi del sangue, terapie concomitanti per evitare interazioni farmacologiche pericolose, eccetera); il secondo serve a etichettare un testo perché sia facilmente elaborato per comporre una prescrizione, abbreviando il tempo perso nella riscrittura di una ricetta ripetitiva, che verrà stampata direttamente con una funzione del menu Stampa.

Non tutti i bollini utilizzati per il tipo di evento e di patologia sono autoevidenti, ma dopo aver consultato qualche volta il manuale, e soprattutto dopo averli

utilizzati molteplici volte, non vi riuscirà difficile ricordarli. Una cosa importante è che tutte le informazioni degli eventi non sono predefinite, pertanto ogni medico rimane libero di codificarle e di raggrupparle secondo i propri schemi mentali senza dover compiere alcuno sforzo di adeguamento a criteri a lui estranei.

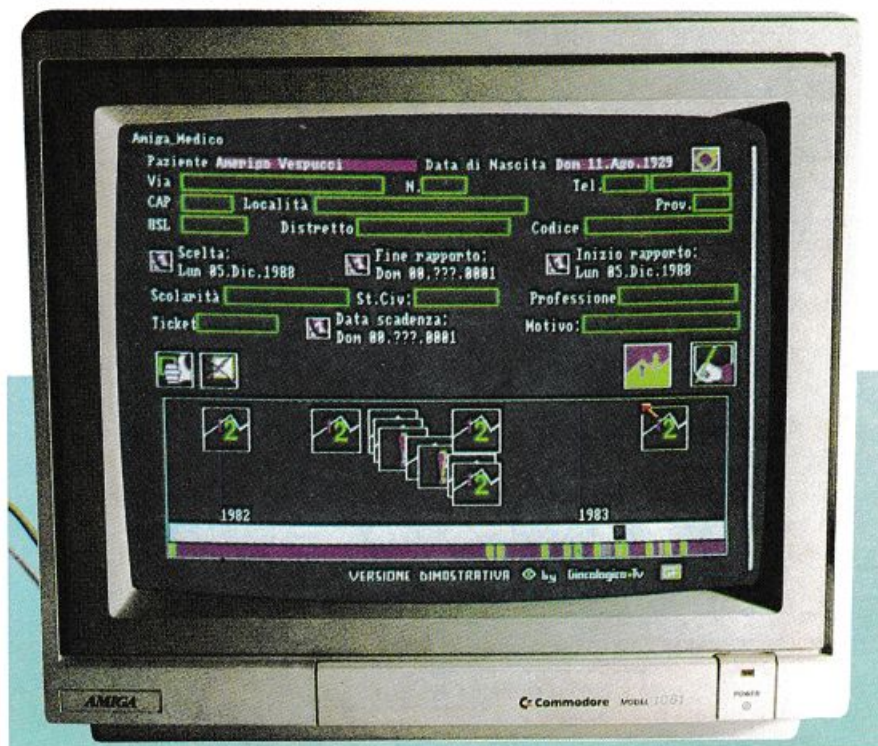
I medici sono considerati proverbialmente incapaci di scrivere in modo leggibile, spesso a ragione, forse perché devono scrivere manualmente una gran quantità di carte, dalle ricette ai promemoria ai pazienti, ai certificati, eccetera.

La difficile lettura delle ricette diventa anche problematica in alcuni casi quando il farmacista o, peggio ancora, il laboratorio d'analisi non riesce a capire quello che c'è scritto. Per ovviare a questi inconvenienti è presente una finestra particolare, richiamabile con la funzione Script, dalla quale si possono attingere,





AMIGA È...



nella compilazione di un testo, tutte le frasi di largo impiego (quelle, per esempio, inerenti agli esami di laboratorio) o di difficile scrittura o memorizzazione (valori normali degli esami, dicitura commerciale corretta di un farmaco, eccetera). Evidentemente la finestra degli Script può essere modificata e personalizzata a proprio piacimento; va ricordato a questo proposito che l'unità di trasferimento è una riga di testo, e che pertanto bisogna definire bene le frasi fatte perché non è possibile inserirne solo una parte nel testo scartando il superfluo, altrimenti l'operazione di eliminazione dovrà essere eseguita sul testo stesso comportando una perdita di tempo.

A volte poi vi potete trovare di fronte a un paziente che presenta numerose patologie concomitanti, e in prima istanza può essere utile visualizzare solo quelle inerenti una certa patologia o meglio ancora una patologia la cui gravità ha richiesto ricoveri ospedalieri; in questo caso Amiga-Medico vi viene in aiuto, for-

nendo con il menù Filtro la possibilità di visualizzare su monitor prima, e di mandare in stampa poi, solo quegli eventi che soddisfano certe condizioni: un certo tipo di evento, una certa patologia, una certa problematica o appunto una combinazione di queste.

La possibilità di stampare i risultati di tutte queste operazioni si rivela importante e utile quando si tratta per esempio di fornire dati relativi alle problematiche del paziente, affinché lo specialista di cui si chiede l'intervento o il medico di accettazione in ospedale ne possano prendere visione e siano aiutati a comprendere meglio il problema del paziente anche qualora questi non si sappia spiegare molto bene, come spesso succede.

I filtri infine sono molto utili quando si desidera eseguire qualche analisi statistica sui propri pazienti, magari per confrontarle con quelle di altri colleghi, e iniziare a svolgere un po' di medicina preventiva sul territorio.

## Conclusione

Il vero problema, quando si analizzano questi pacchetti, non riguarda tanto la capacità delle software house di comporre programmi più o meno potenti nelle funzioni, quanto piuttosto la facilità e la praticità di utilizzo. È infatti una grande perdita di tempo, e fonte di notevole delusione, per un medico dover scrivere prima su carta e poi su computer tutti gli eventi salienti di una visita: generalmente ci si limita alla scrittura su carta. Se quindi un programma, attraverso un'impostazione grafica, permette un utilizzo rapido e intuitivo, converrà preferirlo ad altri, magari più raffinati, ma anche più impegnativi.

L'attenzione da rivolgere al computer potrà essere limitata alla fase finale della visita, durante la quale si scrivono ricette, esami da fare o si forniscono consigli al paziente (quando sono in forma scritta c'è qualche possibilità che non siano immediatamente dimenticati), evitando in questo modo un grosso fastidio al paziente che sente il computer come un intruso che si frappone nel suo rapporto personale col medico. In definitiva si può dire che Amiga-Medico cerca di seguire proprio questa impostazione, e questo fatto rende dunque il programma appetibile a quanti vogliono avvantaggiarsi delle soluzioni informatiche senza intaccare l'essenza primaria della relazione medico-paziente, già così minacciata dall'eccessiva burocrazia e dalla massiccia richiesta di farmaci.

**Giorgio Caspani**

*Amiga-Medico è distribuito da Giocologico, Centro di cultura informatica, via Bolzano 3, Treviso, al prezzo di lire 1.200.000. Amiga con 512 Kb di RAM, due drive 3,5 pollici o hard disk.*





C64

*In questo originale videogame il protagonista è una pallina da tennis che, guidata dal joystick, dovrà superare ostacoli di ogni genere e natura per non rimbalzare sul terreno nemico.*

# Bounder: professione pallina

Cercate di concentrarvi mentre attendete il caricamento di Bounder, poiché vi aspetta un gioco molto divertente che richiede un notevole colpo d'occhio, riflessi pronti e soprattutto una buona capacità di prendere velocemente le decisioni che influiscono in modo determinante sull'esito della partita.

Procedete dunque con il caricamento secondo l'usua-

le procedura: accendete il computer e il monitor, inserite la cassetta allegata alla rivista nel registratore del vostro C64, premete i tasti Ctrl ed Enter piccolo contemporaneamente, premete il tasto Play del registratore e infine date il via all'operazione agendo su un qualunque tasto del computer. Attendete pazientemente il caricamento del programma sfruttando questo intervallo per prepararvi all'azione.

## Che cos'è Bounder?

Non è un omino da guidare in sperduti territori, non è un'astronave con la quale sconfiggere orde di alieni e nemmeno una pistola utile per affrontare una banda di malviventi. Bounder è una pallina da tennis. Poiché non vi abbiamo detto che questo programma è una simulazione del gioco del tennis, vi chiederete: che cosa si può fare con una pallina? È molto semplice: voi non







dovete colpire o lanciare la palla da tennis perché voi siete la palla da tennis. Infatti controllate direttamente Bouncer tramite il joystick o la tastiera senza l'intervento di racchette, mani o accessori vari.

### Il gioco

Vediamo ora un po' che cosa deve fare Bouncer. Descriviamo dapprima la schermata di gioco. Nella parte inferiore dello schermo si trovano alcuni dati: il numero dei salti disponibili per aggiudicarsi i bonus, il numero di vite a disposizione e, naturalmente, il punteggio.

La parte superiore dello schermo è il terreno di gioco. Si tratta di un terreno molto particolare, uno strano posto molto variopinto che raffigura diverse superfici. L'informazione più importante per voi è che potete passare solo sulla superficie grigia a disegni esagonali.

La regola fondamentale per sopravvivere, ovvero per non esplodere e non essere colpiti da corpi acuminati vari, è: se non è grigio evitatelo. In realtà è possibile rimbalzare anche sull'erba, che è raffigurata con un bel verde pieno, ma purtroppo questa appare solo dal settimo schema.

Il movimento di Bouncer è molto semplice: la pallina rimbalza

continuamente mantenendo un ritmo costante.

Voi dovete muovervi mentre la palla è sospesa, facendo attenzione a farla rimbalzare sui quadrati consentiti, che talvolta sono piuttosto lontani. Questo fatto vi costringe a iniziare il movimento esattamente subito dopo il rimbalzo, sfruttando l'intero tempo di volo per effettuare lo spostamento.

Per i vari movimenti potete utilizzare il joystick, oppure i seguenti tasti:

**Q** - per andare a sinistra

**W** - per andare a destra

**P** - per andare in alto, ovvero per procedere lungo il percorso

**L** - per andare in basso, ovvero per retrocedere.

Sia che utilizzate il joystick oppure la tastiera, potete servirvi del tasto Esc per sospendere temporaneamente il gioco mentre, premendo contemporaneamente i tasti Ctrl ed Esc, effettuerete il reset del gioco, ossia ricomincerete da capo.

Per iniziare a giocare premete il tasto Fire sul joystick oppure la barra spazio. Non vi è consentito esitare più di tanto perché il terreno di gioco si muove sempre verso il basso e quindi siete costretti a muovervi in continuazione.

Si possono adottare diverse tecniche di gioco: se state nella parte superiore dello schermo avete molto tempo per riflettere, nel caso siate impegnati in un passaggio difficile, ma non avete modo di accorgervi di eventuali ostacoli che compaiono all'improvviso.

Se vi mantenete nella parte inferiore dello schermo potete vedere molto bene gli ostacoli che vi aspettano, quindi non rischiate di farvi prendere di sorpresa; una qualsiasi esitazione vi costerebbe cara perché non avete la possibilità di retrocedere.

In linea di massima vi possiamo consigliare di rimanere, se è possibile, al centro dello schermo. Chi ha ottimi riflessi può avanzare leggermente, mentre i giocatori che amano meditare le mosse, ma non soffrono di indecisioni paralizzanti, potrebbero ottenere migliori risultati giocando in posizione più arretrata.

### Gli ostacoli

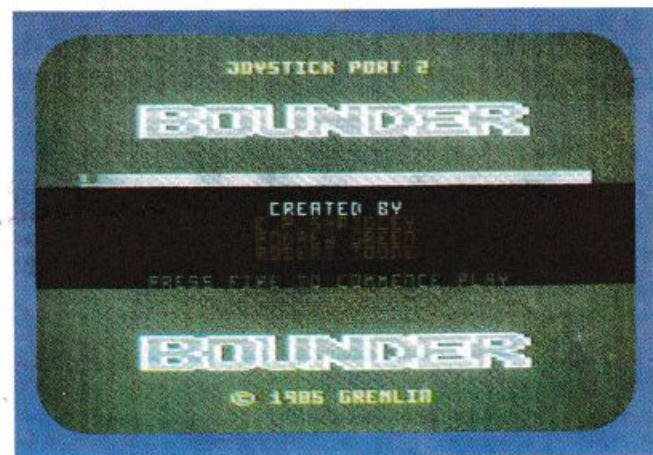
Vediamo ora quali sono gli ostacoli che possono impedire la vostra marcia; ce ne sono di due tipi: fissi o mobili.

#### • Gli ostacoli fissi.

Sono generalmente più facili da evitare: si tratta di muri o montagne che non potete scavalcare con un salto, ma che dovete aggirare. Inutile aggiungere che, se non riuscite a scavalcarli, venite puniti con l'esplosione della pallina. Vengono considerati ostacoli fissi anche i baratri dove cadete se sbagliate il punto di atterraggio della pallina. Tali ostacoli non devono essere aggirati, ma potete scavalcarli: è sufficiente che non ci cadiate dentro.

#### • Gli ostacoli mobili.

Sono ben più pericolosi: ce n'è per tutti i gusti. Iniziamo con le



Schermata introduttiva del gioco con alcuni testi di credits in scrolling.





invasivi zanzare uccello, che procedono sempre controcorrente e vi fanno esplodere semplicemente sfiorandovi.

Non è molto difficile evitarle perché seguono sempre un percorso regolare, anche se si corre il rischio di trovarsele davanti improvvisamente in posti dove non si ha molta possibilità di manovra.

I Binoculoids sono degli strani oggetti di forma vagamente somigliante a un binocolo, che rotolano avanti e indietro in certi punti del percorso. Non è difficile evitarli perché, con un minimo di attenzione, si capisce con quale ritmo si muovono e qual è la zona che coprono e li si può quindi aggirare.

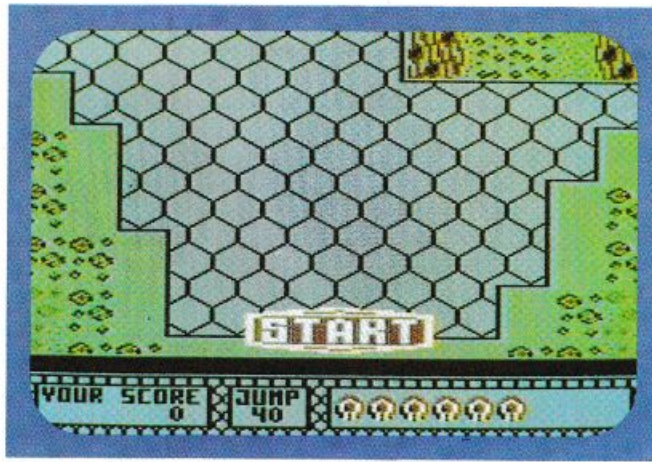
Ancora un insetto: la vespa gigante. Questo vespone è piuttosto insidioso, perché si muove velocemente in linea retta: spesso non fate nemmeno in tempo ad accorgervi della sua presenza, che siete già morti.

Un ostacolo davvero impegnativo è dato dalla scarica elettrica, che ostruisce completamente il passaggio. Per oltrepassare questo punto si deve attendere la fine della scarica, quindi è necessario essere molto pronti per affrontare l'ostacolo con il giusto tempismo.

Si può tentare di passare sopra le antenne che producono la scarica, ma nella maggior parte dei casi la morte è quasi sicura.

Il ventilatore non è una difficoltà insormontabile, anche se solitamente si trova un'intera serie di apparecchi in uno spazio piuttosto ristretto.

L'effetto prodotto dal ventilatore è quantomeno ovvio: si tratta di una continua corrente d'aria che devia la traiettoria di Bouncer; tuttavia, con un minimo di pratica, si impara ad approfittare di tale corrente per eseguire gli spostamenti desiderati.



*Ecco la prima fase del gioco. Per sopravvivere è necessario rimbalzare sugli esagoni in grigio.*

### Gli accessori

Per vostra fortuna il territorio di gioco non è disseminato solamente di ostacoli, in quanto ci sono anche alcuni accessori che semplificano il percorso. Particolare importanza rivestono le caselle con la freccia, che consentono l'esecuzione di un salto molto più lungo del normale. In molti casi l'uso di tali caselle è pressoché essenziale per superare alcuni punti.

Negli schemi superiori possiamo trovare i sistemi di teletrasporto, che ci risparmiano il superamento di barriere invalicabili. Esistono anche altri oggetti utili o nocivi, ma a questo punto pensiamo che sia meglio lasciare a voi il compito di scoprirli, quasi sempre a prezzo di una vita persa.

Un problema diverso si pone per le caselle con il punto interrogativo. Tali caselle sono misteriose, perché possono essere nocive, oppure benefiche. Se siete sfortunati, saltando su una di queste caselle potete venire mangiati da una vorace dentiera, oppure potete essere sfiorati da un'acuminata freccia. Se invece avete fortuna, potete trovare diversi tipi di bonus: per esempio potete vincere dei Jump, ossia dei salti da utilizzare per la prova che si trova alla fine di ogni schema, oppure semplicemente

dei punti. Nel migliore dei casi però, potete anche vincere una o più palline che vi consentono di prolungare la partita.

### Alcune considerazioni

Apriamo una parentesi a proposito delle caselle misteriose per rivelarvi che ogni elemento del videogioco si ripete nel corso di ogni partita sempre allo stesso modo, senza alcuna casualità.

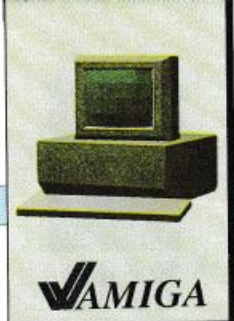
Questo significa, per esempio, che giocando imparate che la prima casella misteriosa è nociva, mentre la seconda no e così via. Anche il movimento dei diversi ostacoli, nonché la posizione degli ostacoli fissi, sono rigorosamente identici in ogni partita.

Se da una parte questa caratteristica di Bouncer rischia di rendere il gioco ripetitivo, è anche vero che in questo modo potete divertirvi a imparare tutte le caratteristiche dell'avventura e quindi potete riuscire ad affrontare tutte le difficoltà.

Un'ultima informazione: al termine di ogni schema di gioco c'è un bonus stage durante il quale dovete cercare di colpire tutti i punti interrogativi nel minor tempo possibile. A questo scopo avete a disposizione un certo numero di salti indicato nella parte inferiore dello schermo.

**Gianni Arioli**





AMIGA È...

# I reconditi perché di AmigaDos

*Continuiamo e concludiamo il discorso sulla strutturazione di un processo multitasking spiegando come avviene la spedizione e la ricezione di dati da una task a un'altra. Inoltre vediamo che cosa si intende per Interruzione, concetto base della programmazione concorrente.*

Le diverse task, oltre a poter sincronizzare il proprio funzionamento utilizzando il sistema delle segnalazioni, devono anche di potersi scambiare dei dati.

Nell'esempio delle due task, visto sul numero scorso, quella che si occupa della tastiera dovrà spedire i dati relativi a tasti premuti alla seconda task, affinché questa possa elaborarli.

Nell'Amiga la spedizione dei dati è affidata al sistema dei messaggi, che consente a una task, che chiameremo trasmittente, di spedire un insieme di dati, detto "messaggio", a un'altra task che chiameremo ricevente.

Il messaggio vero e proprio risulta essere costituito da due distinte parti: un'intestazione, che consente a Exec l'inoltro del messaggio, e il corpo vero e proprio dei dati da trasmettere.

L'intestazione è una struttura che precede il corpo dei dati e che in C si presenta in questo modo:

```
struct Message
{
    struct Node
```

```
mn_Node; /* Nodo di collegamento del messaggio */
    struct MsgPort *mn_ReplyPort; /* Puntatore porta di ritorno */
    UWORD mn_Length; /* Lunghezza corpo dati */
}
```

(La precedente struttura è tratta dal file: include/exec/Ports.h). Come potete vedere, l'intestazione

contiene un nodo (mn\_Node) attraverso il quale questo messaggio potrà essere collegato ad altri messaggi, divenendo così un elemento in una lista di messaggi. La struttura nodo era stata già precedentemente esaminata, insieme alla struttura List, nella scorsa puntata.

La seconda struttura, costituente l'intestazione del messaggio, è un puntatore alla porta di ritorno mn\_ReplyPort (delle porte parleremo fra poco), che può essere immaginato come l'indirizzo del mittente apposto su quel particolare messaggio.

Infine la variabile mn\_Length contiene la lunghezza del corpo dei dati da inviare, che dovrebbe seguire in memoria all'intestazione.

Il sistema dei messaggi permette alla task trasmittente di inviare un messaggio indipendentemente dal fatto che la task ricevente sia pronto a riceverlo. Infatti quest'ultima task potrebbe essere ancora impegnata nell'elaborazione di dati inviati precedentemente, e sarebbe poco





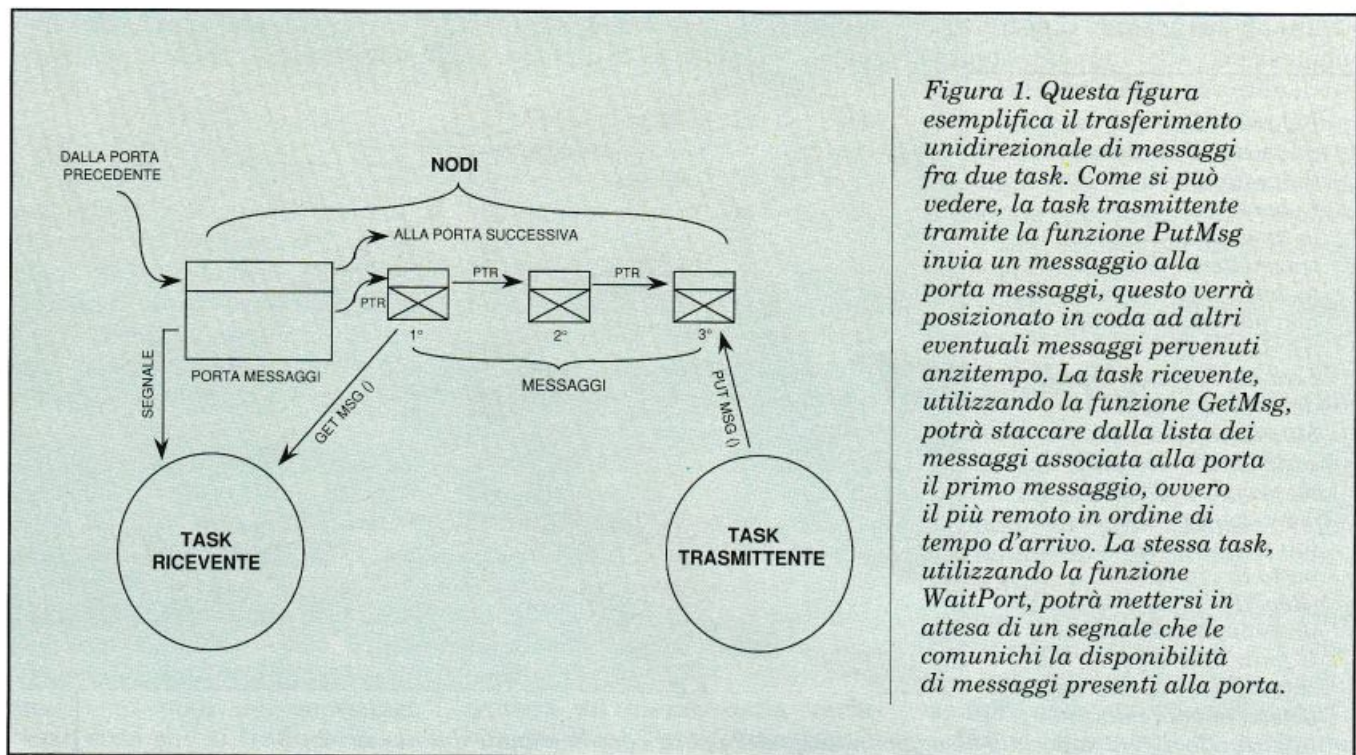


Figura 1. Questa figura esemplifica il trasferimento unidirezionale di messaggi fra due task. Come si può vedere, la task trasmittente tramite la funzione PutMsg invia un messaggio alla porta messaggi, questo verrà posizionato in coda ad altri eventuali messaggi pervenuti anzitempo. La task ricevente, utilizzando la funzione GetMsg, potrà staccare dalla lista dei messaggi associata alla porta il primo messaggio, ovvero il più remoto in ordine di tempo d'arrivo. La stessa task, utilizzando la funzione WaitPort, potrà mettersi in attesa di un segnale che le comunichi la disponibilità di messaggi presenti alla porta.

efficiente far attendere la task trasmittente.

Per evitare questo inconveniente il messaggio non viene immediatamente sottoposto all'attenzione della task ricevente, ma viene invece collegato a una lista di altri messaggi diretti alla stessa task, da dove potrà essere prelevato a tempo debito.

### Il sistema delle porte messaggio

Il sistema operativo, e in particolare la libreria Exec, mette a disposizione delle Porte Messaggi, alle quali potranno essere inviati i vari messaggi affinché il sistema operativo provveda all'inoltro automatico dei medesimi.

Ecco qui di seguito la struttura di queste porte (tratta dal file: include/exec/ports.h):

```
struct MsgPort
{
    struct Node mp_Node; /*
    Nodo di collegamento porta */
    UBYTE mp_Flags; /*
```

```
Stato della porta */
    UBYTE mp_SigBit; /*
    Segnale da inviare al ricevimen-
    to del messaggio */
    struct Task *mp_Task; /*
    Puntatore alla task a cui inviare
    il segnale */
    struct List mp_MsgList;
    /* Lista dei messaggi arrivati alla
    porta */
}
```

Anche questa struttura, come le precedenti, inizia con un Nodo, attraverso il quale la struttura MsgPort viene collegata a una lista di sistema, ovvero viene gestita direttamente da Exec, di nome PortList.

La variabile mp\_Flags conterrà invece dei bit di stato, i quali indicheranno come si dovrà comportare la porta ogni qualvolta riceverà un nuovo messaggio.

Le variabili mp\_SigBit e \*mp\_Task verranno prese in considerazione più avanti, quando si parlerà della sincronizzazione fra la porta messaggi e la task ricevente.

Infine alla struttura mp\_MsgList farà capo la catena degli eventuali messaggi pervenuti alla porta.

Per poter utilizzare una Porta per l'inoltro di un messaggio, occorrerà innanzitutto crearla, sempre che essa non sia già presente nella omonima lista di sistema.

Allo scopo la libreria Exec mette a disposizione la funzione AddPort, la quale come parametro d'ingresso necessita di un puntatore alla struttura MsgPort. Tale funzione provvede a inizializzare la sopraddetta struttura, e a collegarla alla lista PortList.

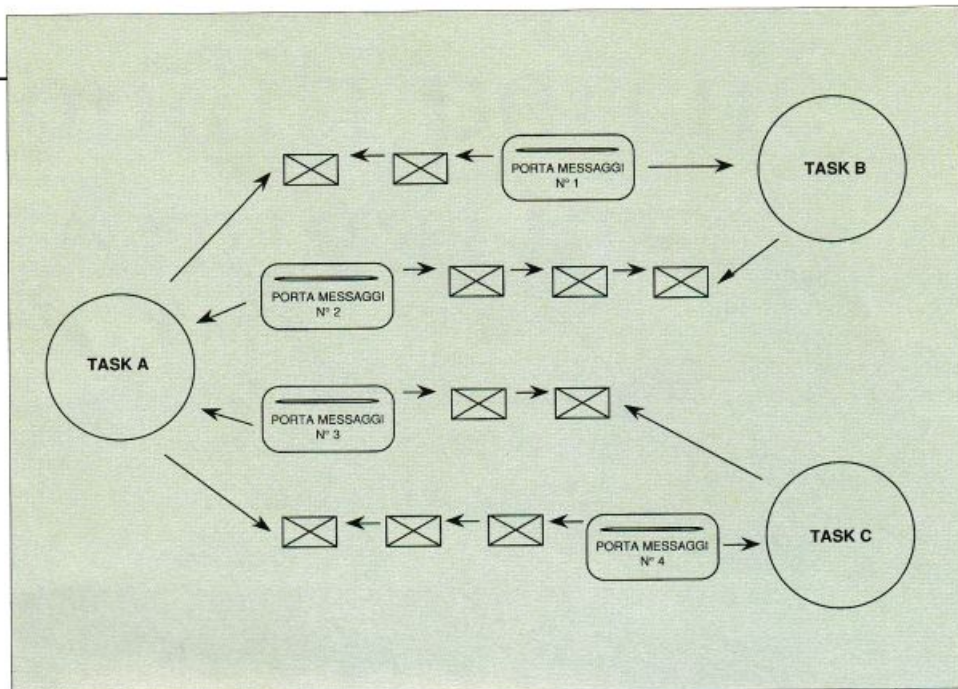
Da questo momento la nuova porta sarà a disposizione come magazzino postale per l'inoltro di qualsiasi messaggio.

Reciproca della funzione AddPort è RemPort, la quale provvede a rimuovere una porta dalla lista delle porte operative. Tale funzione, analogamente alla precedente, si aspetta come parametro d'ingresso un puntatore alla porta da rimuovere.





Figura 2. Esempificazione di un possibile flusso di messaggi fra tre task. Come si può vedere, la task A può ricevere e trasmettere messaggi alle task B e C, che a loro volta non possono comunicare fra di loro direttamente. I canali di trasferimento dei messaggi sono bidirezionali, in quanto costituiti da due porte messaggio orientate in versi opposti. Ognuna di tali porte costituirà una porta di ripetizione (ReplyPort) per i messaggi provenienti alla task tramite l'altra porta. Lo stesso flusso di messaggi poteva essere realizzato



utilizzando un'unica porta messaggio in luogo delle porte contrassegnate dai numeri 2 e 3. In questo caso a tale porta sarebbero stati inviati i messaggi provenienti sia dalla task B sia dalla C. Quest'ultima soluzione sarebbe la migliore nel caso in cui la task A debba porsi contemporaneamente in attesa di messaggi provenienti sia dalla task B sia dalla task C, dato che la funzione WaitPort, da usarsi in questi casi, permette di porsi in attesa di messaggi rispetto a un'unica porta.

Infine la funzione FindPort consente di determinare l'ubicazione, ovvero la posizione in memoria della corrispondente struttura, di una qualsiasi porta, conoscendone il nome, puntato dalla variabile \*In\_Name nel nodo della struttura relativa alla porta stessa.

### Spedizione di un messaggio

Per spedire un messaggio, la task trasmittente dovrà allocare nella propria memoria lo spazio necessario alla struttura Message coi dati che si intende inviare.

Sucessivamente la stessa task, tramite la funzione PutMsg, invierà il messaggio alla porta.

Questa funzione, che necessita come parametri i puntatori alla porta e al messaggio, provvederà a inserire quest'ultimo in coda al-

la lista di messaggi che faranno capo alla porta in questione.

Detta operazione sarà portata a termine agendo esclusivamente sui puntatori situati nel nodo del messaggio stesso (mp\_Node), al fine di collegarlo agli altri messaggi, eventualmente presenti nella lista facente capo a mp\_MsgList nella struttura relativa alla porta.

Tutti i messaggi che verranno mandati a quella porta, utilizzando la funzione PutMsg, verranno ad accrescere, nello stesso ordine di arrivo, la lista facente capo alla porta stessa.

### Ricezione di un messaggio

La task ricevente, terminato lo svolgimento della precedente elaborazione, potrà controllare l'arrivo di eventuali messaggi

utilizzando la funzione GetMsg che, aspettandosi come parametro d'ingresso un puntatore alla porta che potrebbe contenere un eventuale messaggio, fornirà, come parametro di ritorno, un puntatore al primo messaggio disponibile in tale porta, oppure il valore 0 nel caso la porta sia vuota ovvero non contenga alcun messaggio.

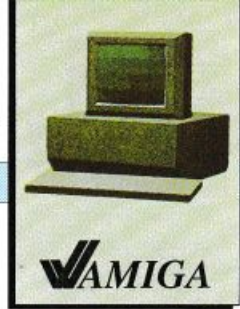
La funzione stessa provvederà allo scollegamento del primo messaggio disponibile, sempre che ve ne sia uno, dalla lista facente capo alla porta.

### Caratteristiche delle porte messaggi

Da quanto detto emerge il fatto che le porte messaggio costituiscono una sorta di buffer (deposito) temporale per i messaggi.

Infatti i messaggi verranno





prelevati dalla task ricevente, nei tempi che essa stessa deciderà e nello stesso ordine cronologico in cui erano stati spediti dalle varie task trasmettenti.

In secondo luogo la porta messaggi costituisce un canale unidirezionale di trasporto dei messaggi, in quanto questi ultimi procederanno da una o più task trasmettenti in direzione di un'unica task ricevente, ma non potranno in nessun caso procedere nel verso opposto.

Infine il sistema dei messaggi si rivela particolarmente efficiente in quanto i singoli messaggi non verranno trasportati fisicamente attraverso la memoria, ma il loro spostamento sarà semplicemente dovuto alla variazione dei puntatori contenuti nei nodi dei messaggi stessi, cosa che si traduce in efficienza e velocità.

### Porte di ripetizione

A questo punto però ci si può chiedere come possa la task trasmettente sapere che il messaggio è stato inoltrato, ovvero ricevuto dalla task ricevente.

Infatti, come abbiamo visto, la trasmissione e la ricezione di un messaggio, a causa della presenza delle porte messaggio, non sono da considerarsi eventi simultanei.

Pertanto la task trasmettente non dovrà modificare il proprio messaggio, né tantomeno disallocarlo, fin quando non sarà accertato che la task ricevente lo abbia effettivamente letto.

Il sistema utilizzato all'uopo da Exec consiste nel rimandare al mittente il messaggio tramite un'altra porta messaggi, detta porta di ripetizione (Reply Port).

In questo modo, quando una task trasmettente avrà ricevuto indietro il messaggio, potrà essere certa dell'avvenuta ricezione di questo da parte della task rice-

vente e potrà riutilizzare tale messaggio, o la memoria a esso associata, per altri scopi.

Nulla impedisce alla task ricevente di inserire un'eventuale risposta nel campo dati del messaggio prima di ritornarlo alla task trasmettente.

Il messaggio di ritorno potrebbe essere inoltrato utilizzando un'altra porta, detta porta di ripetizione (reply), tramite la funzione PutMsg, oppure tramite la più specifica ReplyMsg, la quale come parametro d'ingresso si aspetta un puntatore al messaggio che si desidera venga rinviato al mittente.

La funzione stessa provvederà a chiamare PutMsg, passandole come parametro relativo alla porta d'inoltro l'indirizzo che troverà all'interno della variabile ms\_ReplyPort del messaggio stesso.

### Attesa di messaggi

Il sistema dei messaggi e delle relative porte, come è stato visto sinora, funzionerebbe certamente ma avrebbe difetto analogo a quello presentato dai segnali prima che prendessimo in considerazione lo stato di attesa e la relativa funzione Wait.

Infatti anche in questo caso la task ricevente, terminato l'ultimo compito a essa affidato, dovrà interrogare periodicamente la porta, o le porte nel caso siano più di una, dalle quali attende un eventuale messaggio.

Tutto il tempo utilizzato in questa attesa attiva sarà sottratto alle altre task concorrenti.

La libreria Exec, per ovviare a questo inconveniente, mette a disposizione la funzione WaitPort la quale necessita, come parametro d'ingresso, un puntatore alla porta dalla quale si attende un messaggio.

La funzione stessa controllerà lo stato della porta per verificare

l'eventuale presenza di messaggi. Se qualche messaggio dovesse essere collegato alla lista interna alla porta, il controllo verrà restituito alla procedura chiamante che, utilizzando la funzione GetMsg, potrà scollegare tale messaggio dalla lista e leggerlo.

Viceversa, se nessun messaggio risultasse essere presente nella lista della porta, la funzione WaitPort alloccherà un segnale fra quelli disponibili nella task ricevente (utilizzando la funzione AllocSig), e scriverà il numero di tale segnale nella variabile: mp\_SigBit della struttura MsgPort.

Sucessivamente scriverà l'indirizzo della stessa task ricevente nel puntatore: \*mp\_Task, sempre nella struttura MsgPort.

Infine chiamerà la funzione Wait per quel particolare segnale allocato, onde far sì che la task, entrando in stato di attesa, venga temporaneamente sospesa dall'esecuzione in multitask.

Quando più tardi una task trasmettente chiamerà la funzione PutMsg per inviare un messaggio alla porta in questione, questa stessa funzione, automaticamente, utilizzando la funzione Signal, invierà il segnale contenuto nella variabile mp\_SigBit alla task il cui indirizzo sarà specificato nel puntatore mp\_Task, provocando così il termine dello stato di attesa in cui tale task (la ricevente) si trovava.

Quando più tardi, dopo le necessarie commutazioni da parte della gestione del multitasking, tale task ricevente riprenderà il controllo del microprocessore, potrà, utilizzando la funzione GetMsg, scollegare il messaggio dalla porta e leggerlo.

Vediamo ora i sistemi delle interruzioni, delle librerie e dei dispositivi, cercando di dare un'idea della filosofia di base che ha ispirato Exec.





## Le interruzioni

Tutti i calcolatori dispongono di un meccanismo di gestione delle interruzioni, ovvero di un insieme di dispositivi hardware che permettano l'interruzione del programma in esecuzione e consentano l'avvio di una particolare procedura, al presentarsi di un determinato segnale hardware.

Il sistema delle interruzioni fu inizialmente concepito per migliorare la gestione delle operazioni di Input/Output. Poniamo per esempio che un calcolatore ipotetico, che non disponga di un particolare meccanismo di gestione delle interruzioni, debba ricevere dei dati da una porta seriale.

Il microprocessore di tale calcolatore dovrà periodicamente interrogare i registri dell'interfaccia a cui farà capo la porta seriale, per controllare l'arrivo di un eventuale dato.

Questa operazione comporterà una diminuzione dell'efficienza del sistema, imputabile a due ragioni: primo, il microprocessore dovrà perdere un certo quantitativo di tempo per controllare i re-

gistri dell'interfaccia.

Secondo: il programma in esecuzione dovrà contenere nel proprio codice delle istruzioni che prevedano il controllo periodico dei registri ubicati nell'interfaccia.

Utilizzando il sistema delle interruzioni sarà invece la stessa interfaccia ad avvertire il microprocessore della disponibilità di dati da elaborare, abbassando una particolare linea elettrica detta richiesta di interruzione (IRQ).

All'abbassamento di questa linea, il microprocessore risponderà salvando nello stack il contenuto del contatore istruzioni (Program Counter), e del registro di stato (Status Register), per poi mandare in esecuzione una particolare routine, la quale si occuperà del reperimento del dato disponibile sulla porta seriale e della sua eventuale elaborazione.

Al termine di questa procedura il controllo del flusso di elaborazione verrà restituito al programma che era stato interrotto, il quale riprenderà la propria esecuzione esattamente nel pun-

to in cui questa era stata sospesa, all'arrivo del segnale d'interruzione.

## Il sistema delle interruzioni in Amiga

Il meccanismo delle interruzioni, così come è presente nei calcolatori a processo singolo, è anche utilizzato nei sistemi multitask quale l'Amiga.

Infatti in questo calcolatore esistono diversi dispositivi hardware, che possono originare segnali di richiesta di interruzione.

Per l'esattezza esistono 14 diversi segnali di richiesta di interruzione, i quali vengono raggruppati dal Chip "Paula" in 6 classi di priorità, alle quali corrisponderà una richiesta d'interruzione che verrà inoltrata al microprocessore 68000.

Maggiore è il livello di priorità associato a una interruzione, minore è la probabilità che la procedura d'interruzione venga interrotta dal verificarsi di un'ulteriore interruzione. Infatti il 68000 accetterà solo interruzioni il cui valore di priorità sia maggiore di quella in corso di esecuzione.

### Tavola 1.

Livello	Num.Int.	Descrizione evento
6	13	Segnale da espansione esterna
5	12	Byte disponibile dal disco
5	11	Buffer ricevente porta seriale UART pieno
4	10	Canale audio 3 DMA (termine esecuzione stringa campionata)
4	9	Canale audio 2 DMA
4	8	Canale audio 1 DMA
4	7	Canale audio 0 DMA
3	6	Blitter DMA (termine operazione blitter)
3	5	Fine scansione schermo
3	4	Coprocessore grafico
2	3	Porte di I/O: tastiera, stampante, espansione
1	2	Interruzione software (originata da funzione Cause())
1	1	Termine trasferimento blocco dati da disco
1	0	Buffer trasmittente porta seriale UART pieno



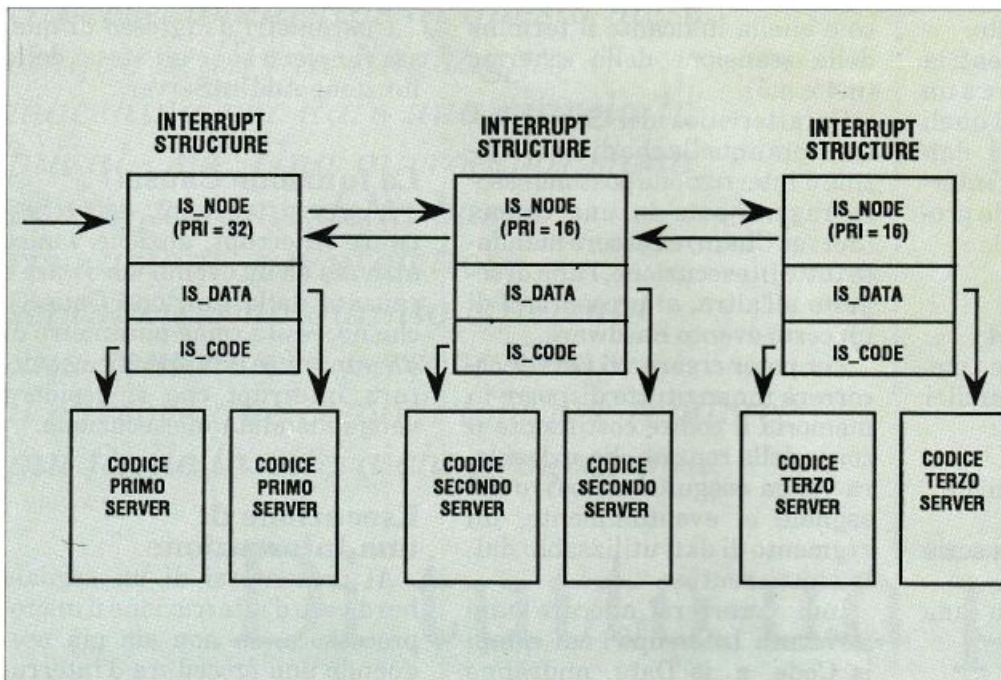


Figura 3.  
Collegamento di  
vari Server in una  
Server Chain.  
Durante  
l'esecuzione  
dell'interruzione  
asservita da questa  
Server Chain,  
l'intera lista di  
strutture "Interrupt"  
verrà scansionata e  
le relative procedure  
verranno eseguite,  
in ordine  
decescente di  
priorità, come  
rispecchiato  
dall'ordine stesso  
della lista.

La **tavola 1** riproduce la tabella indicante tutti i possibili dispositivi generanti interruzioni e il livello di priorità a essi associati. Come si può vedere, il segnale d'interruzione non mascherabile NMI a livello di priorità 7 non è utilizzato all'interno dell'Amiga.

### La gestione delle interruzioni

Nell'Amiga, la gestione delle interruzioni da parte del sistema operativo è più complessa di quanto non lo sia in un calcolatore a processo singolo; basti considerare che tutto il sistema di commutazione fra le task è controllato da procedure che avvengono in stato d'interruzione.

In secondo luogo, il fatto che nel calcolatore esistano diverse task concorrenti, suggerisce che debba anche essere possibile associare diverse procedure d'interruzione a ogni singolo evento hardware. Poniamo per esempio che due task attive desiderino veder eseguita una loro propria routine ogni qualvolta venga premuto un tasto sulla tastiera.

In questo caso sarà compito del sistema operativo, e per l'esattezza di Exec, di mandare in esecuzione, una di seguito all'altra, entrambe le routine associate alle due task.

Inoltre il sistema delle interruzioni dovrà essere congnato in maniera tale che ogni singola routine d'interruzione sia trasparente all'altra, ovvero in modo che le due procedure, eseguite in interruzione, possano ignorarsi a vicenda.

Il sistema operativo dell'Amiga, tramite Exec, consente 2 distinti modi di operare con le interruzioni. Nel primo, a una possibile sorgente hardware di interruzione viene associata una specifica routine d'interruzione detta Interrupt Handler, che verrà mandata in esecuzione al presentarsi di quel particolare segnale hardware.

Nel secondo modo, al dispositivo verrà associata un'intera lista di routine d'interruzione Interrupt Server, le quali verranno eseguite una di seguito all'altra al verificarsi della interruzione stessa.

### Struttura Interrupt

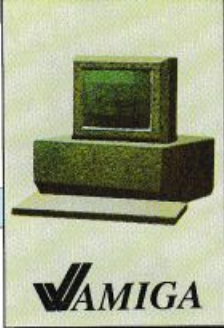
Ogni routine d'interruzione, indipendentemente dal fatto che si tratti di un Handler o di un Server, per poter essere eseguita necessiterà della presenza in memoria della specifica struttura detta Interrupt, a essa associata.

Ecco l'aspetto di tale struttura tratto dal file "Include/exec/Interrupts.h":

```
struct Interrupt
{
    struct node is_Node; /*
    nodo di collegamento */
    APTR is_Data; /* puntatore
    al segmento dati dell'interruzione */
    VOID (*is_Code)(); /*
    puntatore codice dell'interruzione */
};
```

Come si può vedere, la struttura risulta essere costituita da un nodo is\_Node, che permetterà, nel caso si tratti di un Server, di collegare questa specifica routine d'interruzione alle altre che verranno eseguite al mani-





festarsi di uno stesso evento.

Gli altri elementi costituenti la struttura sono un puntatore a un segmento di dati `is_Data` (i quali potranno essere utilizzati dall'interno della procedura d'interruzione), e un puntatore alla procedura stessa: `(*is_Code)()`.

### Gli Handler Interrupt

Le interruzioni gestite mediante il sistema degli Handler Interrupt sono le seguenti:

- quelle relative al disco (n.1 e n.12)
- quelle relative alla porta seriale UART (n.0 e n.11)
- quelle relative ai quattro canali audio (n.7,8,9 e 10)
- quella del blitter (n.6)
- le interruzioni software attivate tramite la funzione `Cause()` (n.2).

Pertanto a ognuna di queste possibili sorgenti d'interruzione sarà possibile associare un'unica routine.

Per poter impostare una routine come Handler Interrupt, la libreria Exec mette a disposizione la funzione `SetIntVector()` che, come parametri d'ingresso, necessita del numero relativo all'interruzione alla quale si intende associare la routine, e di un puntatore alla struttura Interrupt relativa alla routine stessa.

La funzione `SetIntVector()` ritornerà come parametro di ritorno un puntatore alla struttura Interrupt precedentemente associata a quella specifica interruzione.

### I Server Interrupt

Le interruzioni gestite mediante il sistema dei Server sono le seguenti: quelle provenienti dalle eventuali espansioni (n. 3 e n. 13), quella del coprocessore grafi-

co e quella indicante il termine della scansione dello schermo (n.4 e n.5).

Caratteristica dei Server Interrupt è quella che diverse routine d'interruzione possono essere raggruppate in una catena (Server Chain) ed essere mandate tutte in esecuzione, l'una di seguito all'altra, al presentarsi di un certo evento hardware.

Per poter creare un Server occorrerà innanzitutto disporre in memoria il codice costituente il corpo della routine che si desidera venga eseguita all'arrivo del segnale e, eventualmente, un segmento di dati utilizzabili dalla stessa routine.

Indi occorrerà allocare una struttura Interrupt i cui campi `is_Code` e `is_Data` andranno compilati rispettivamente con dei puntatori al codice e ai dati della routine d'interruzione già presente in memoria.

Occorrerà anche inizializzare la variabile `ln_Pri`, presente nel nodo della struttura, a un valore compreso fra i seguenti: -32, -16, 0, 16, 32.

Tale numero indicherà la priorità relativa di quel determinato Server, ovvero la sua posizione rispetto a tutti gli altri Server presenti nella lista relativa a quella interruzione. Più alta sarà la priorità, prima verrà eseguito quel particolare Server.

Infine, utilizzando la funzione `AddIntServer()`, si collegherà la struttura Interrupt alla lista facente capo a quel particolare segnale hardware.

La funzione `AddIntServer()` si aspetta gli stessi parametri di `SetIntVector()`, ovvero il numero di interruzione cui si desidera associare la routine e il puntatore alla struttura Interrupt.

Reciproca di questa funzione è `RemIntServer()`, la quale rimuove un determinato Server dalla lista associata a un'interruzione.

I parametri d'ingresso di questa funzione sono gli stessi della funzione `AddIntServer`.

### La funzione Cause()

L'interruzione n.2, detta Software Interrupt, anziché venire attivata da un evento hardware è causata dalla funzione `Cause()`, che necessita come parametro di un puntatore alla specifica struttura Interrupt che si desidera venga mandata in esecuzione.

### Esecuzione di una Interruzione

Al presentarsi di un segnale hardware d'interruzione il microprocessore, se non sia già eseguendo una procedura d'interruzione di livello pari o superiore, entrerà in stato di Eccezione, ovvero abiliterà lo stato Supervisor, e salverà sullo stack di sistema il contenuto del contatore d'istruzione (PC) e del registro di stato (SR). Successivamente il controllo del flusso verrà passato a Exec, che provvederà al salvataggio di tutti i registri del microprocessore sullo stack supervisor (non quello relativo alle singole task), e manderà in esecuzione l'handler o il primo server associati a quella particolare fonte di interruzione.

Se quel particolare interrupt è asservito da una catena di Server, terminata l'esecuzione del primo server il controllo tornerà a Exec, il quale a sua volta lo passerà al Server successivo, onde eseguire l'intera catena.

Al termine dell'ultimo Server, Exec ripristinerà i registri del microprocessore ai valori precedenti l'interruzione, e tornerà il controllo alla procedura task o a un'altra interruzione di priorità inferiore che era stata precedentemente interrotta.

**Sergio Fiorentini**  
(continua)





C64

*Gli eccezionali tips di questo mese: cinque utility in linguaggio macchina per dare una marcia in più al C64. Una microespansione per la gestione della bassa risoluzione, un protettore semplicissimo quanto efficiente, una routine per personalizzare il cursore due comodi tool per rendere più facile la vita a chi programma.*

## Di tutto un po'

I listati di questo mese sono un po' più lunghi del solito.

Avendo presente la difficoltà di digitazione dei listati, che contengono parecchie istruzioni

Data, abbiamo pensato di fare cosa gradita ai nostri lettori pubblicando, sulla cassetta allegata, i cinque mini programmi di questa rubrica.

### Spursor

Personalizzare il proprio computer è una costante aspirazione di ogni programmatore. Chiunque abbia un minimo di esperien-

### Listato 1. Spursor

```

10 rem -----
20 rem -
40 rem - spursor -
50 rem -
60 rem -----
70 :
72 fort=1 to 8*25:reada$:q=asc(left$(a$,1)):w=asc(right$(a$,1))
74 if q>57 then q=q-55:goto 77
75 q=q-48
77 if w>57 then w=w-55:goto 80
78 w=w-48
80 poke 827+t,16*q+w:next
82 fort=1 to 64:reada$:q=asc(left$(a$,1)):w=asc(right$(a$,1))
84 if q>57 then q=q-55:goto 87
85 q=q-48
87 if w>57 then w=w-55:goto 90
88 w=w-48
90 poke 703+t,16*q+w:next
100 data 78,a9,60,a0,03,8d,14,03
110 data 8c,15,03,58,20,44,e5,a9
120 data 10,a2,2b,8d,0e,d0,8c,0f
130 data d0,a9,0b,8d,ff,07,a9,80
140 data 8d,15,d0,60,a5,d3,a4,d6
150 data c9,46,10,0f,c9,28,10,10
160 data c9,1e,10,07,c9,00,10,08
170 data 4c,81,03,a2,80,4c,7e,03
180 data a2,00,8e,10,d0,be,e4,03
190 data 8e,0f,d0,a8,b9,94,03,8d
200 data 0e,d0,20,ea,ff,4c,61,ea
210 data 10,18,20,28,30,38,40,48
220 data 50,58,60,68,70,78,80,88
230 data 90,98,a0,a8,b0,b8,c0,c8
240 data d0,d8,e0,e8,f0,f8,00,08
250 data 10,18,20,28,30,38,40,48
260 data 10,18,20,28,30,38,40,48
270 data 50,58,60,68,70,78,80,88
280 data 90,98,a0,a8,b0,b8,c0,c8
290 data d0,d8,e0,e8,f0,f8,00,08
300 data 10,18,20,28,30,38,40,48
310 data 2b,33,3b,43,4b,53,5b,63
320 data 6b,73,7b,83,8b,93,9b,a3
330 data ab,b3,bb,c3,cb,d3,db,e3
340 data eb,00,00,00,4e,01,0d,05
341 :
342 rem
343 rem dati sprite
344 rem
345 :
350 data 00,00,00,00,00,00,00,00
360 data 00,00,00,00,00,00,00,01
370 data ff,80,01,00,80,01,00,80
380 data 01,00,80,01,00,80,01,00
390 data 80,01,00,80,01,00,80,01
400 data 00,80,01,00,80,01,ff,80
410 data 00,00,00,00,00,00,00,00
420 data 00,00,00,00,00,00,00,00
430 :
440 rem *****
445 rem **
450 rem * per attivare spursor: *
460 rem **
470 rem * sys 828*
480 rem **
490 rem *****

```





**C64**

## TIPS & TRICKS

### Listato 2. Magic Screen

```
1 rem -----
2 rem -
3 rem - magic screen -
4 rem -
5 rem -----
155 sa= 49152
160 :
165 hb=int((sa+11)/256)
170 lb=(sa+11)-hb*256
175 :
190 :
195 poke sa,162:poke sa+1,lb
200 poke sa+2,160:poke sa+3,hb
205 fori=sa+4to sa+35:readx:pokei,x:next
210 data142,8,3,140,9,3,96
215 rem-
220 data 32,115,0,201,33,240,6
225 data 32,121,0,76,231,167,32,115,0,32,36,192,32,121,0,76,231,167
230 :
240 :
245 s=0:for i=sa+36 to sa+217
250 readx:pokei,x:s=s+x
255 next
260 if s<>23004 thenprint"[clr]errore[32]nei[32]data":stop
265 data216,32,250,174,32,158,183,134,89,32,241,183,134,90,32,241,183,134
270 data 91,32,241,183,134,92,32,247,174,32,241,183,134,253,32,241,183,134
275 data254,165,90,56,229,89,24,105,1,133,252,165,92,24,105,1,133,92
280 data 56,229,91,133,251,169,128,133,93,173,136,2,133,88,162,0,134,87
285 data166,89,240,16,165,87,24,105,40,133,87,165,88,105,0,133,88,202
290 data208,240,173,17,208,42,144,250,169,11,141,17,208,184,166,253,164,91
295 data177,87,80,26,145,87,200,196,92,208,245,165,87,24,105,40,133,87
300 data165,88,105,0,133,88,198,252,208,226,80,38,177,87,224,0,240,14
305 data224,1,240,20,224,2,240,20,224,3,240,8,208,18,36,93,240,6
310 data208,8,165,254,80,200,9,128,80,196,41,127,80,192,169,27,141,17
315 data208,96
```

#### NOTE:

Per attivare il programma digitare sys sa.  
Sintassi: !(vz,bz,vs,bs),f,c  
vz = prima linea  
bz = ultima linea  
vs = prima colonna  
bs = ultima colonna  
f= funzione (0-3)  
0= inverte  
1= reverse  
2= normale  
3= riempie  
c= codice di riempimento (=codice video)

za di programmazione in linguaggio macchina o in Basic avrà sicuramente tentato di sviluppare diversi tool per rendere più confortevole e piacevole una fase particolare della programmazione. Spursor rientra in questa categoria di utility: permette di sostituire il normale cursore con uno sprite e quindi di personalizzarne la forma e le dimensioni.

Per usare Spursor dovete copiare il **listato 1** e dare il consueto Run. L'attivazione avviene, come indicato nelle linee Rem poste alla fine del listato, con sys 828. Potete personalizzare il cursore

secondo i vostri gusti facilmente: la definizione dello sprite che sostituisce il cursore è memorizzata nel blocco numero 11, cioè nella zona di memoria compresa fra gli indirizzi 704 e 767. Pur non essendo rilocabile, la routine dovrebbe essere facilmente utilizzabile nei vostri programmi perché viene allocata in una zona di memoria usata raramente: il buffer di cassetta.

### Magic screen

Magic screen è una microespansione utilissima che vi aiu-

terà a creare incredibili effetti grafici in bassa risoluzione. Questa utility vi permette infatti di trattare porzioni rettangolari del video alla velocità del linguaggio macchina per riempirle di caratteri, invertirne il contenuto o farle lampeggiare.

Per utilizzare Magic screen dovete copiare il **listato 2**, dare il Run e quindi digitare: sys sa. A questo punto avete a disposizione un nuovo comando Basic: !(vz, bz, vs, bs), f, c. I primi quattro parametri, che vanno sempre racchiusi nelle parentesi tonde, specificano la zona rettangolare dello schermo su cui si vuole agire. Il parametro f indica il tipo di operazione che si vuole compiere. Ecco i valori che si possono usare:

**0:** inverte il contenuto della porzione di schermo.

**1:** pone in reverse il contenuto della porzione di schermo.

**2:** pone in modo normale il contenuto della porzione di schermo.

**3:** riempie la porzione rettangolare con un carattere.

L'ultimo parametro c rappresenta invece il codice video del carattere con cui si vuole riempire la porzione di schermo selezionata.

Questo parametro va sempre indicato, ma viene utilizzato soltanto se il valore del parametro f è 3.

### Protector plus

La protezione del software è il problema fondamentale, e per ora senza soluzione, di chi produce software commerciale.

Le tecniche sviluppate fino a





### Listato 3. Protector plus

```

10 rem -----
15 rem - -
20 rem - protector plus -
25 rem - -
30 rem -----
60 :
65 :
100 poke53280,6:poke53281,6:poke646,14
110 print"[clr]"
120 fori=0to41:read a:poke828+i,a:next
130 sys828
140 fori=52992to53061:read a
150 pokei,a:next
160 input"[down][down][32][32]code[32]";c$:c=len(c$)
170 fori=ltoc
180 poke53072+i,asc(mid$(c$,i,1)):next
190 poke62954,32:poke62955,00:poke62956,207
200 poke62626,32:poke62627,00:poke62628,207
210 poke62731,76:poke62732,12:poke62733,207
220 poke63019,76:poke63020,30:poke63021,207
230 poke13*16^3-1,c:poke1,53:new
240 :
250 :
260 data 169,0,133,95,133,90,133,88,169
270 data 160,133,96,169,192,133,91,133,89,32
280 data 191,163,169,0,133,95,133,90,133
290 data 88,169,224,133,96,169,0,133,91,133,89
300 data 76,191,163
310 :
320 :
330 data 142,253,207,162,1,142,254
340 data 207,174,253,207,96,32,46,207
350 data 141,252,207,56,173,253,207
360 data 237,252,207,164,147,76,14
370 data 245,141,252,207,32,46,207
380 data 24,109,252,207,32,221,237
390 data 76,46,246,206,254,207,208
400 data 6,173,255,207,141,254,207
410 data 142,253,207,174,254,207,189
420 data 80,207,174,253,207,96
421 :
430 rem *****
440 rem * *
450 rem * poke 1,53 per codificare*
460 rem * *
470 rem * poke 1,55 per decodificare *
480 rem * *
490 rem *****

```

questo momento non hanno dato buoni frutti: anche la protezione migliore riesce a malapena a resistere qualche ora agli assalti di un pirata abbastanza abile. Una possibile soluzione sta nella crittografia.

Protector plus è in grado di crittografare qualunque programma Basic utilizzando un codice numerico unico per ogni programma. Il risultato di questa operazione è un programma perfettamente copiabile ma assolutamente inutilizzabile se prima non viene decrittato (sempre con Protector plus).

Per utilizzare Protector plus dovete copiare il **listato 3** e dare il Run.

Vi verrà chiesto di inserire il codice segreto di protezione, cioè il codice usato per crittografare il programma. Più lungo è il codice segreto maggiori saranno le probabilità di difendere il proprio programma.

In generale è consigliabile utilizzare codici abbastanza corti e facili da ricordare perché, se dovesse andare perso il foglio su cui avete scritto il codice, avreste ancora qualche probabilità di recuperare il programma.

Scelto il codice dovete digitare Poke 1,53 e quindi salvare il programma con la solita istruzione Save.

Per caricare correttamente il programma trattato in questo modo dovete lanciare nuova-

### Listato 4. Fine Listing

```

0 adr = 49152
1 rem -----
2 rem - -
3 rem - fine listing -
4 rem - -
5 rem -----
6 :
10 fori=adr to adr+87
20 read a:s=s+a:poke i,a
30 next
31 if s<>10407thenprint"[clr]errore[32]nei[32]data":end
36 hi=int((adr+11)/256)
37 lo=adr+11-int((adr+11)/256)*256
38 poke adr+6,hi
39 poke adr+1,lo
40 sys adr:poke 2,0:end
50 rem -----
60 rem per attivare fine listing :
70 rem poke 2,0
80 rem -----
90 rem per disattivare fine listing:
99 rem poke 2,<>0
100 rem -----
101
data169,11,141,6,3,169,48,141,7,3,96,16,3,76,28,167,201,58,240,3,76,243
102 data166,166,2,240,3,76,243,166,169,13,32,210,255,152,72,160,2,177,95
103 data170,200,177,95,133,98,134,99,162,144,56,32,73,188,32,223,189,162
104 data0,189,0,1,240,3,232,208,248,169,32,32,210,255,202,208,250,104,168
105 data169,32,76,243,166,48,141,7,7,96

```





**C64**

## TIPS & TRICKS

mente Protector plus, inserire il codice segreto del programma da caricare e quindi procedere al caricamento del programma.

### Fine listing

La lacuna più evidente del Basic del C64 è la totale assenza di istruzioni rivolte al debugging dei programmi. Per testare correttamente anche una semplice routine di poche decine di righe si rischia di perdere diverse ore con il rischio poi di non concludere nulla di buono.

Per controllare bene un programma che superi di molto le dimensioni dello schermo è assolutamente indispensabile effettuare una stampa del listato. Non è possibile infatti ottenere a

video un listing abbastanza chiaro e completo che permetta immediatamente di cogliere l'errore.

Fine listing è una preziosissima utility che vi permetterà finalmente di effettuare comodamente il debugging dei vostri programmi direttamente al video. Fine listing modifica la routine di list dell'interprete rendendo la visualizzazione delle linee di programma estremamente più chiara.

Tutti gli statements separati dal simbolo dei due punti (:) che compaiono sulla stessa linea di programma verranno visualizzati su linee diverse. In tal modo riuscirete molto facilmente a seguire il flusso del programma anche senza ricorrere alla stam-

pa del listato.

Per utilizzare Fine listing dovete copiare il **listato 4** e dare il Run. La routine si installa da sola e quindi, quando compare il ready del Basic, siete già pronti per usare Fine listing.

Per poter correggere una linea dovete innanzitutto disabilitare Fine listing con Poke 2,1, listare la linea e quindi effettuare la modifica come di consueto. Per riattivare Fine listing dovete usare l'istruzione Poke 2,0.

### Escape

L'editor del C64 presenta una sola lacuna veramente insopportabile: il quote mode. Questa modalità operativa viene attivata ogni volta che si digita il carattere dei doppi apici (") oppure il carattere Insert e serve per consentire l'inserimento di caratteri speciali (movimenti del cursore, cancellazione schermo, colori, eccetera) nelle stringhe. Quando è attivo il quote mode si perde gran parte della libertà di movimento perché il cursore non risponde più correttamente ai tasti di movimento. La soluzione del problema è Escape. Questa routine vi permetterà di uscire dal quote mode con la semplice pressione di un tasto e senza dare luogo a effetti collaterali.

Per utilizzare Escape dovete copiare il **listato 5**, dare il Run e quando compare il ready del Basic digitare sys 49152. Quando vi trovate nel quote mode e volete uscirne non dovete far altro che premere il tasto funzione F7. L'effetto è praticamente equivalente a quello che si ottiene premendo contemporaneamente i tasti Shift e Return con il vantaggio che il cursore non va a capo. La routine non è rilocabile e per disattivarla dovete premere contemporaneamente i tasti Run/Stop e Restore.

**Daniele Maggio**

### Listato 5. Escape

```

10 rem -----
20 rem - - -
30 rem - escape -
40 rem - - -
50 rem -----
60 :
72 fort=1 to 8*18:reada$:q=asc(left$(a$,1)):w=asc(right$(a$,1))
74 if q>57 then q=q-55:goto 77
75 q=q-48
77 if w>57 then w=w-55:goto 80
78 w=w-48
80 poke 49151+t,16*q+w:next
100 data 78,a9,11,a2,c0,8d,14,03
110 data 8e,15,03,58,a9,02,85,fb
120 data 60,c6,fb,f0,03,4c,31,ea
130 data a9,02,85,fb,a9,00,85,fc
140 data a5,d4,f0,11,a9,ff,85,fc
150 data a9,11,8d,27,04,a9,01,8d
160 data 27,d8,4c,3a,c0,a9,20,8d
170 data 27,04,a5,c7,f0,0d,a9,12
180 data 8d,26,04,a9,01,8d,26,d8
190 data 4c,50,c0,a9,20,8d,26,04
200 data a5,d8,f0,11,a9,ff,85,fc
210 data a9,09,8d,25,04,a9,01,8d
220 data 25,d8,4c,6a,c0,a9,20,8d
230 data 25,04,a5,d7,c9,88,d0,1b
240 data a2,00,86,d4,86,c7,86,d8
250 data a5,fc,f0,0f,e8,86,c6,a4
260 data d3,88,a5,ce,91,d1,a9,9d
270 data 8d,77,02,4c,31,ea,00,a0
300 :
310 rem *****
320 rem *
330 rem * per attivare:
335 rem *
340 rem * sys 49152
350 rem *
360 rem * per uscire dal quote mode:
370 rem *
380 rem * f7
390 rem *
400 rem *****

```





*Vi siete mai chiesti come il famoso Rob Hubbard componesse le stupende colonne sonore dei migliori videogames? O come certi sofisticati programmi potessero elaborare agilmente grafica e musica contemporaneamente e indipendentemente? Vorreste poter fare anche voi tutto questo?*

# Fiat sonus! E suono fu

Ebbene, con Composer è possibile. Questa fantastica utility permette di accessoriare i vostri programmi con musiche di sottofondo. Il bello è che i programmi in Basic o in linguaggio macchina non dovranno occuparsi minimamente di gestire la musica che funziona in interrupt. L'interrupt è una particolare tecnica di programmazione che consente l'elaborazione di un programma secondario contemporaneamente a quello principale e indipendentemente da questo. Potete comporre brani su otto ottave a tre voci, definire forme d'onda e involuppi, gestire i filtri e la risonanza, sistemare in memoria, dove volete, i brani composti e attivarli con una semplice Sys. Il tutto occupa, alla fine, solo pochi byte, proprio come i file musicali dei programmatori professionisti (10 blocchi di programma suonano per più di un minuto).

## Come iniziare

Caricate il programma Composer dal disco su cui l'avete trasferito servendovi dell'utility Dsave V2 che trovate sulla cassetta subito dopo il programma introduttivo.

Al termine del caricamento comparirà sul video la schermata riportata in **figu-**





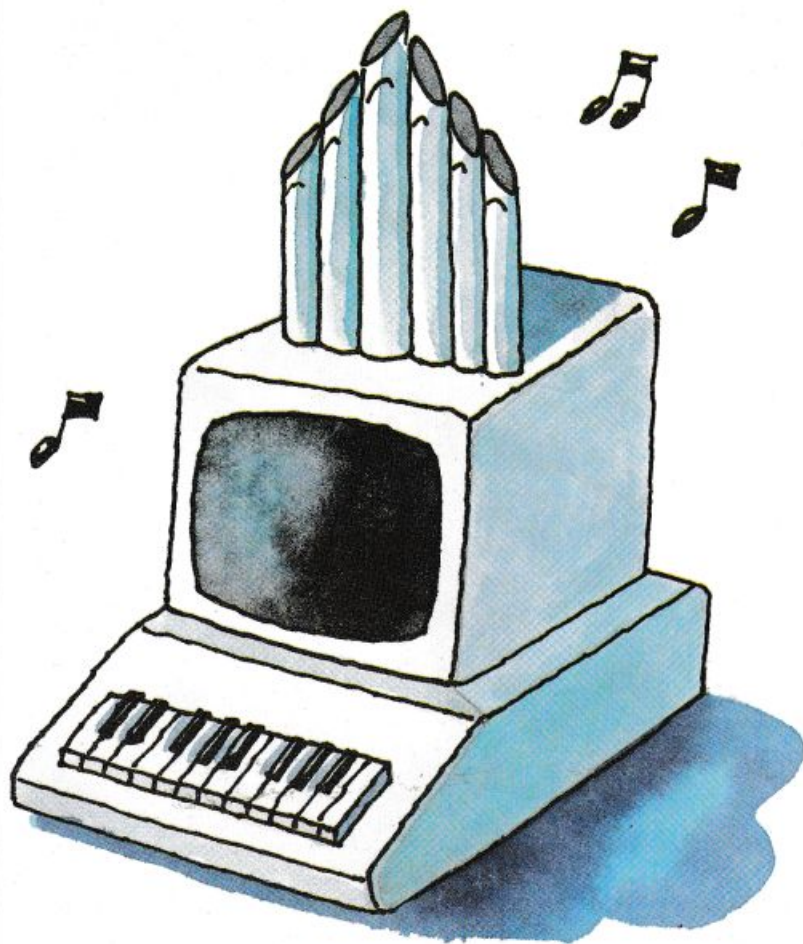


**C64**

## UTILITY

**ra 1.** Sulla sinistra del video sono elencate le principali opzioni del programma tutte associate a un tasto funzione del vostro C64. La parte destra del video mostra quello che potremmo definire lo spartito su cui disporremo le note, anche se in verità questo spartito non ha pentagramma e le note sono delle lettere. Inizialmente lo spartito è vuoto. Nella parte alta del video vedete, sulla sinistra, un piccolo quadro che informa sulla funzione correntemente attivata, mentre sulla destra è riportato un promemoria sulle note a cui sono associati i tasti inferiori della tastiera in fase di composizione del brano.

Ora vediamo come si usa il programma spiegando punto per punto quali operazioni effettuare per ottenere un brano.



### • L'esempio.

Appena caricato il programma la memoria riservata ai dati del brano è occupata da una simpatica musichetta d'esempio. Se premete il tasto F7, infatti, potete udirne l'esecuzione. Come sicuramente avrete notato la melodia si sviluppa su tre voci. Lo spartito, infatti, presenta tre colonne (V-1, V-2, V-3) su cui vengono disposte verticalmente le note di ogni singola voce.

Per vedere lo spartito del brano d'esempio dovete premere il tasto funzione F2 che recupera dalla memoria i dati e li dispone sullo spartito. Mediante i tasti cursore potete visionare tutto il brano. La prima colonna dello spartito (Line) indica il numero della battuta. Ad ogni battuta possono essere suonate tre voci contemporaneamente. La seconda colonna dello spartito indica il tempo per ogni battuta.

### • Un nuovo brano.

Lo spartito e la memoria dati non sono in diretto collegamento. In pratica, la melodia eseguita premendo il tasto F7 è quella in memoria, non quella sullo spartito.

Premendo il tasto F1 si imposta la memoria dati secondo il contenuto dello spartito. Per questa ragione se effettuate delle modifiche sullo spartito non potrete riscontrarle nel brano in esecuzione fino a che non avrete premuto F1.

Per comporre un nuovo brano dovete innanzitutto cancellare lo spartito. Questa operazione si effettua premendo contemporaneamente Shift e Clr/Home. A questo punto il cursore individua la prima battuta dello spartito, vuota. Supponiamo di stabilire, come nell'esempio, un tempo di battuta fisso, di 6. Usando i tasti cursore posizionatevi sulla seconda colonna dello spartito e digitate 6. Ora il cursore si trova sulla colonna della prima voce invitandovi ad introdurre il primo dato. Continuando con l'esempio, supponiamo di voler scrivere una scala cromatica di note sulla terza ottava (il C64 ne può suonare 8). Digitate dunque il tasto 3 (terza ottava) e il tasto Z. Il tasto Z, come ricordato dal promemoria in alto a destra sul video, corrisponde alla nota Do. Sullo spartito, però, comparirà il simbo-



lo C, che indica la nota Do nella notazione anglosassone.

A questo punto il cursore dovrebbe trovarsi sulla colonna della seconda voce. Noi, però, vogliamo programmare una scala di note solo sulla prima voce. Per questa ragione premete il tasto Return. Ora il cursore è sulla seconda battuta, alla colonna Tempo. Digitate ancora 6. Quando il cursore è sulla colonna della prima voce digitate 3 e poi battete il tasto S. Sullo spartito compare il simbolo C#. Nella notazione anglosassone questo segno significa Do diesis.

Il promemoria in alto a destra non indica quali tasti sul computer rappresentano i diesis, ma la disposizione dei tasti neri del promemoria corrisponde a quella dei tasti del computer. Il Re#, ad esempio, si trova tra il Re e il Mi, cioè tra i tasti X e C, ovvero il tasto D. Proseguite nella composizione della scala cromatica fino alla nota Si.

#### • Funzioni di editing.

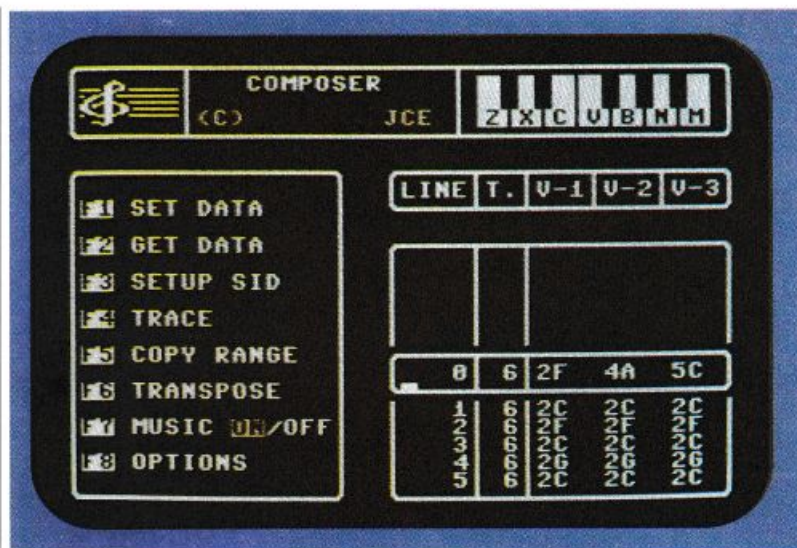
Durante la digitazione dello spartito è possibile usufruire di alcune funzioni di editing. A parte la possibilità di cancellare lo spartito intero con Shift e Clr/Home, è possibile cancellare la linea di spartito su cui si trova il cursore premendo Inst/Del. Per aggiungere una linea di spartito occorre invece premere Inst/Del contemporaneamente a Shift.

#### • Un accorgimento.

Sicuramente avrete notato che la digitazione delle note risulta difficoltosa perché ai tasti premuti sul computer non corrispondono i simboli sullo spartito. Per sopperire a questo inconveniente vi suggeriamo di scrivere con una matita sui tasti del computer i simboli delle note anglosassoni corrispondenti. Se i tasti del vostro computer sono troppo scuri potete ricorrere alla soluzione illustrata in **figura 2**. Con un pezzo di nastro adesivo trasparente fissate su ogni tasto un quadratino di carta su cui avrete riportato i simboli opportuni.

#### • Ascoltare e perfezionare il brano.

Se premete il tasto F7 udite ancora il brano d'esempio. Infatti la memoria non è ancora stata informata con i dati del



nuovo spartito. Premete dunque il tasto F1 e poi il tasto F7. La vostra scala cromatica viene eseguita continuamente. Sul brano in esecuzione è possibile effettuare alcune importanti modifiche in tempo reale.

Premete il tasto F3. La schermata che compare è riprodotta in **figura 3**. Nella parte alta del video, sulla destra, sono riportati i tasti corrispondenti alle opzioni disponibili da questa sezione del programma:

- *Inviluppo*. Il primo pannello di controllo (il quadro Sid #1) permette di modificare in tempo reale l'inviluppo delle tre voci in esecuzione. Nel nostro esempio viene suonata solo la voce 1 (colonna V-1). Usando i tasti cursore evidenziate il quadro che volete modificare e premete i tasti F5, F6, F7 o F8. Provate a disporre i valori 9 per Attack e Decay e 0 per Sustain e Release.

- *Forma d'onda*. Il secondo pannello (in basso a sinistra) permette di cambiare, sempre in tempo reale, la forma d'onda. Il Sid del C64 può emettere quattro tipi di forma d'onda. Provatele tutte disponendovi con il cursore sui quadri Wave e premendo F7. La forma d'onda Pulsar (Pul) richiede la definizione di un ulteriore parametro. La sezione Pulsar Wide (P. Wide) sullo stesso pannello permette la specificazione di questa caratteristica. Sullo stesso pannello trovate la sezione Filter.

*Figura 1.*  
La schermata principale del programma. Da questa sezione è possibile editare lo spartito e gestire alcune importanti funzioni.





C64E

## UTILITY

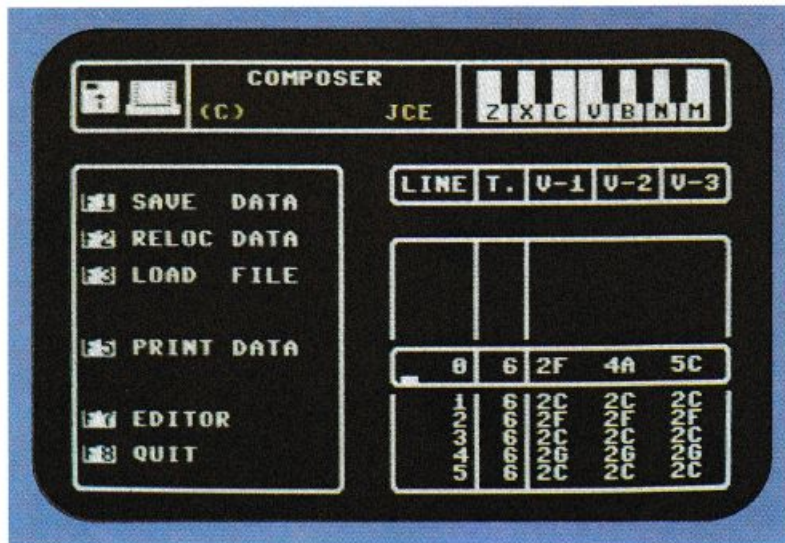
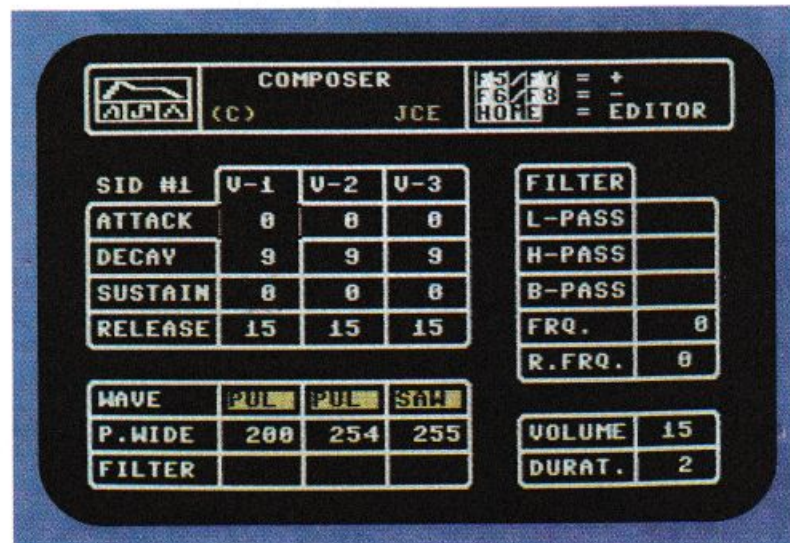


Figura 3.  
Il menù options.  
Questa schermata  
compare premendo  
il tasto F8 dal  
menù principale.

Figura 2.  
La possibilità  
di mutare  
le caratteristiche  
del suono in tempo  
reale costituisce  
l'eccellenza  
di questa utility.

Ogni quadro di questa sezione è un Toggle, ovvero un interruttore che attiva o disattiva il filtro per la voce a cui si riferisce.

- *I filtri.* I filtri consentono di eliminare alcune frequenze del suono piuttosto che altre. Sul pannello Filter sono riportati i tipi di filtri che è possibile porre per le tre voci. Sullo stesso pannello trovate la Frequenza (Frq) di riferimento per il filtro selezionato e la Risonanza di Frequenza (R. Frq). La risonanza accentua l'effetto del filtro attivato. Piuttosto che conoscere la teoria sui filtri è molto più utile ed economico (in termini di spazio) effettuare molte prove a caso fino a che non si in-



tuisce la dinamica e il comportamento di questo dispositivo. Provate a selezionare la forma d'onda Noise dal secondo pannello e il filtro Low Pass dal pannello Filter con una frequenza di 200 e una risonanza di frequenza di 15. Per attivare il filtro sulla voce 1 ricordate che dovete mettere a On il toggle relativo nel secondo pannello.

- *Altri parametri.* L'ultimo pannello (in basso a destra) consente di modificare il volume globale della melodia e il tempo globale della stessa (duration).

Premendo il tasto Home tornate alla schermata principale del programma.

### Le opzioni disponibili

A questo punto non ci resta che effettuare una panoramica su tutte le opzioni di questa stupenda utility spiegando le funzioni non sperimentate con gli esempi.

#### • Menù principale.

Il menù di figura 1 riporta le seguenti opzioni:

*F1 - Set Data.* Trasferisce i dati della melodia dallo spartito alla memoria dati.  
*F2 - Get Data.* Trasferisce i dati della melodia dalla memoria dati allo spartito.  
*F3 - Setup Sid.* Conduce alla schermata di figura 3 per le modifiche in tempo reale della melodia.

*F4 - Trace.* Mentre la melodia è in esecuzione è possibile vedere, in tempo reale, quale battuta viene eseguita. Questa funzione è indispensabile per individuare eventuali battute da modificare.

*F5 - Copy Range.* In fase di editazione dello spartito può essere utile copiare intere porzioni della melodia. Attivate questa opzione dopo esservi posizionati alla prima battuta della porzione da copiare, evidenziate la zona con i tasti cursore, premete Return, posizionatevi all'inizio della zona destinazione e premete nuovamente Return.

*F6 - Transpose.* Questa opzione consente di cambiare la chiave di lettura di tutto il brano eseguendo uno spostamento di tonalità di tutte le battute della melodia





C64E

secondo uno spostamento campione. Premete F6, digitate la nota originale (per esempio 3C), battete Return, digitate la nota in cui lo spostamento trasforma quella appena digitata (per esempio 4C, spostamento di un'ottava verso l'alto) e battete nuovamente Return. Nel caso d'esempio, tutte le note dello spartito risulteranno più acute di un ottava.

**F7 - Music On/Off.** Attiva e disattiva l'esecuzione continua della melodia.

**F8 - Options.** Attiva un menù secondario per le funzioni di input/output.

#### • Menù Options.

Questo menù viene attivato premendo il tasto F8 dal menù principale, vediamo le opzioni:

**F1 - Save Data.** La melodia in memoria viene salvata su disco sotto forma di un file programma, allocato a partire dalla locazione di memoria 49152 (\$C000 esadecimale). Per suonare la melodia da un programma Basic basta effettuare una SYS 49152 e la musica farà da sottofon-

do al vostro programma senza interferire minimamente col programma stesso. Affinché tutto fili liscio caricate il file della melodia prima del vostro programma, in modo da evitare un messaggio di out of memory.

**F2 - Reloc Data.** Con questa opzione è possibile allocare a piacere i dati della melodia. Il file scaricato su disco in questo modo può essere caricato e lanciato con una SYS alla locazione specificata prima del salvataggio.

**F3 - Load File.** Il file dati salvato su disco può essere richiamato con questa opzione.

**F5 - Print Data.** Se possedete una stampante potete mettere su carta il contenuto dello spartito. L'opzione consente di stabilire da quale battuta iniziare la stampa.

**F7 - Editor.** Questo tasto riconduce al menù principale di **figura 1**.

**F8 - Quit.** Termina il programma con un falso reset. Per rientrare nel programma digitate SYS 64738.

**Raffaele Zanini**

## Per trasferire i programmi di RE&C da cassetta a disco

Su ogni numero di Radio Elettronica & Computer trovate l'utility Dsave V2, registrata subito dopo il file d'intestazione. Questo programma consente di trasferire la maggior parte dei programmi della cassetta su un floppy disk. Gli unici programmi che non è possibile trasferire sono quelli che, per ragioni di copyright, sono salvati sulla cassetta in modo autostart.

Il programma Dsave V2 è d'uso semplicissimo: dopo il caricamento del solito menù all'inizio della cassetta si deve caricare e lanciare Dsave v2 con LOAD"DSAVE\*". Il menù che compare sul video offre tre possibilità:

1 - La cassetta verrà letta e il primo programma incontrato caricato. A questo punto viene chiesta conferma per il trasferimento sul disco, dopodiché si passerà al caricamento del successivo programma sulla cassetta e così via.

2 - Scegliendo la seconda opzione, invece, verranno salvati su disco tutti i programmi automaticamente, senza selezioni da parte dell'utente. A questo proposito bisogna ricordare che, a volte, sulla cassetta vi sono dei file sequenziali o programmi particolari che il copiatore non può trasferire, nel qual caso il processo si blocca. Vi consigliamo quindi di utilizzare l'opzione 1 anche se volete trasferire tutti i programmi della cassetta.

3 - Questa opzione consente di visionare la directory del disco.





C64

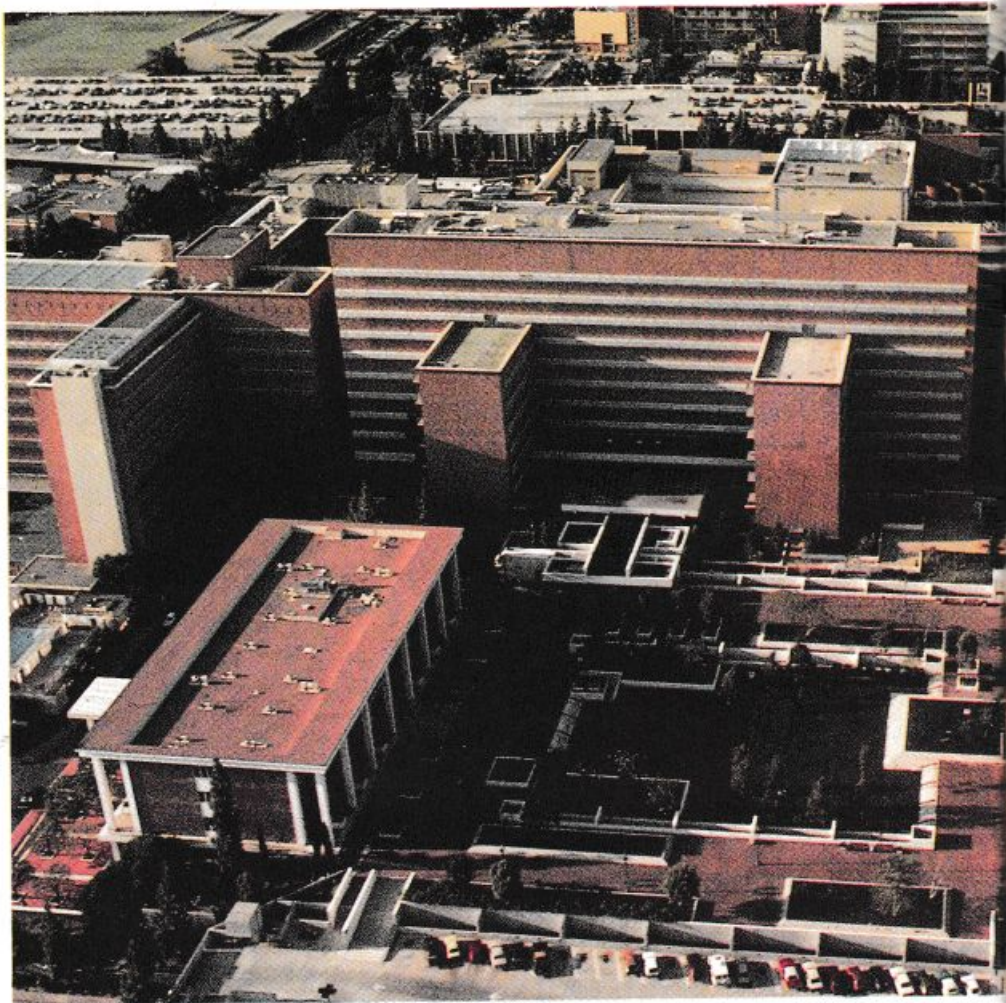
CONTABILITÀ

# Un minuto per districare la matassa

*Da oggi si può calcolare il prezzo di qualsiasi affitto abitativo in meno di un minuto, senza possibilità di errore e considerando tutti i parametri che entrano in gioco in particolari situazioni, come nelle locazioni di appartamenti a struttura complessa, con eventuali autorimesse o cantine o con parti condominali, su diversi livelli di piano: questo per tutte le zone d'Italia.*

Il programma che apre questa nuova rubrica volta a risolvere i più svariati problemi di contabilità con l'aiuto del Commodore 64, contiene al suo interno complesse routine di elaborazione che permettono di determinare, con la massima semplicità, l'ammontare base annuo (in lire) dell'equo canone relativo a qualsiasi contratto di locazione a uso abitativo posto in essere secondo quanto prevede la legge numero 392 del 1978 (unitamente alle modificazioni a questa apportate negli anni seguenti).

Il patrimonio abitativo italiano ammonta a quasi 18 milioni di unità immobiliari, e di queste oltre il 30% è concesso in affitto. È noto a chiunque sia proprietario di un immobile locato (anche soltanto un piccolo appartamento) quanto complicato e laborioso sia interpretare e applicare correttamente detta legge, nata per regolare il mercato degli affitti (e per poterne determinare gli equi importi), ma rivelatasi subito una disciplina abordabile unicamente da esperti del settore come i geometri, che esigono sem-

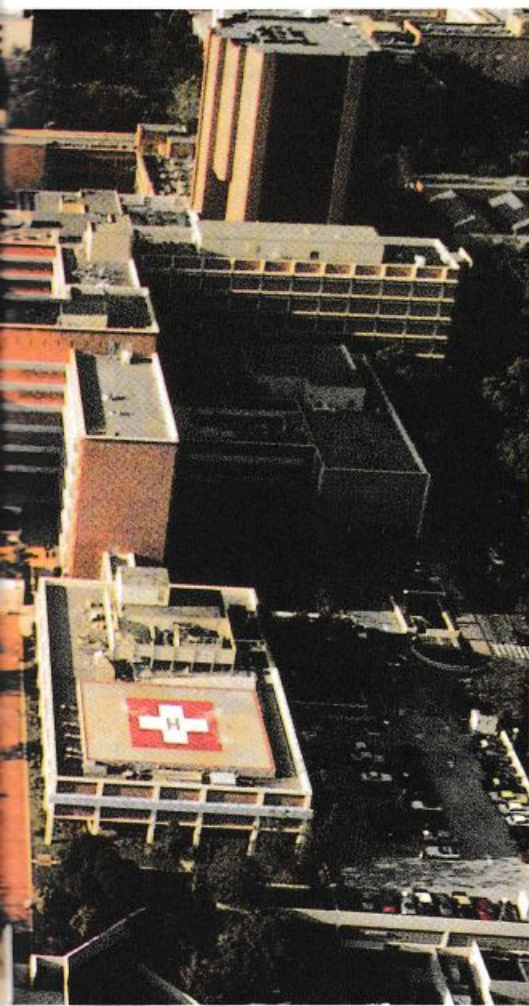






pre, da chi chiede consulenza, noiose precisazioni, giorni di tempo e sostanziose parcelle per mettersi a lavorare su metri quadrati, coefficienti e comma legislativi.

Col software Equo Canone 1 si può calcolare il prezzo di qualsiasi affitto abitativo in meno di un minuto, senza possibilità di errore e considerando tutti i parametri che entrano in gioco in particolari situazioni, come nella locazione di appartamenti a struttura complessa, con eventuali autorimesse o cantine o con parti condominiali, su diversi livelli di piano: questo per tutte le zone d'Italia (soggette a diverso trattamento in base a collocazione geografica, demografia, tensione abitativa).



Si tratta insomma di un utile strumento di lavoro ideale non solo per geometri, stimatori, ingegneri e agenzie immobiliari, ma anche per qualsiasi persona interessata a conoscere le esatte procedure pratiche di determinazione legale del canone.

All'utente viene semplicemente chiesto di specificare o indicare, tramite i tasti numerici o di funzione, valori di superfici e categorie di selezione degli appositi coefficienti di misurazione: su questi input il computer lavora velocemente, calcolando i vari importi parziali e fornendo alla fine ben sei videate con i totali di Superficie reale Netta e Convenzionale, di Costo unitario di produzione, di Valore locativo e di Canone annuo di base (da cui è possibile poi ricavare direttamente l'ammontare dell'equo canone periodico semplicemente aggiungendo le maggiorazioni previste dalla legge, come l'aggiornamento Istat).

È importante ricordare che il programma ha validità per i soli contratti di locazione a uso di abitazione, e sono quindi esclusi tutti gli altri (ad esempio le locazioni di fabbricati industriali o di negozi), anche perché è la legge stessa a cui si fa riferimento che disciplina l'equo canone unicamente per detta destinazione di immobili.

Tramite un'unica pagina grafica multicolore si può controllare con facilità lo svolgimento di tutte le routine di elaborazione, con monitoraggio effettuato da un multidisplay che permette una lettura istantanea e soprattutto grande facilità di interpretazione: il tutto senza possibilità di errori.

Il programma, che si trova sulla cassetta allegata alla rivista col nome "Equo Canone 1" può essere facilmente trasferito su disco servendosi dell'utilità Dsave V2, anch'essa sulla cassetta. Il

caricamento da cassetta si effettua avvolgendo completamente il nastro, digitando LOAD"EQUO CANONE 1" seguito dalla pressione del tasto Return, digitando infine, al termine del caricamento, il comando Run seguito ancora da Return. Il caricamento dal disco avviene in modo analogo, digitando, però, LOAD"EQUO CANONE 1",8,1. Vediamo ora come utilizzare il programma.

### Funzionamento e uso

Terminata l'operazione di caricamento, trascorsi pochi secondi da quando il programma viene fatto partire (tramite un Run di lancio) compare la pagina-monitor a fondo e bordo di colore nero, che rimane invariata nella sua parte grafica e strutturale durante tutte le fasi di esecuzione delle routine.

Nella zona superiore dello schermo sono presenti la grande scritta "Equo Canone 1" (nei colori giallo-viola e bianco-verde) e il data set di copyright (in colore grigio reverse). Un sottotitolo in arancio reverse evidenzia la funzione del software, che è quella di permettere un'immediata definizione dell'ammontare del canone annuo di base relativo a qualsiasi tipo di contratto di affitto a uso abitativo (stipulato tra locatore e conduttore).

Nella parte inferiore dello schermo, evidenziata da una cornice gialla, trova sistemazione il multidisplay per la scrittura dei messaggi-guida, per gli input e per la proiezione delle sei videate finali con i risultati.

Contenuti e riferimenti legislativi del programma, nonché procedure di input ed elaborazione dei dati, fanno riferimento alle disposizioni della legge, detta dell'equo canone, n. 392 del 27/07/1978 (in particolare agli artt. dal 12 al 24 di questa) e successive modificazioni, nonché natu-





ralmente alla vastissima legislazione vigente in materia di contratti di locazione, proroga degli sfratti e rilevazioni di indici statistici Istat.

Anche se non esplicitamente precisato su video, si intende che l'uso del programma "Equo Canone 1" non comporta, comunque e da chiunque venga usato, alcuna responsabilità per chi lo ha realizzato, prodotto e pubblicato: occorrerà pertanto controllare sempre attentamente ogni operazione che porta poi al calcolo dell'importo del canone annuo di base (risultato finale).

La scritta iniziale lampeggiante in vari colori avverte che la routine esecutiva degli input può essere avviata solo premendo il tasto Return, in caso contrario il computer rimane in stand by.

Si inizia subito con la richiesta dei dati che permettono di definire la superficie dell'immobile: per questo motivo bisognerà avere già disponibili i valori di ampiezza di tutti i vani, compresi eventualmente i posti macchina, la superficie scoperta in godimento esclusivo, la quota d e l

verde condominiale, nonché tutte le altre metrature anche indirettamente assimilabili all'immobile stesso oggetto di analisi.

A ogni immissione un'apposita maschera di input fornisce tutte le informazioni necessarie affinché l'utente dia risposte adeguate: in rosa e azzurro viene stampato il testo della richiesta e più sotto due indicatori delimitano un campo in cui viene poi visualizzato il valore specificato, che può avere un'ampiezza massima di ben 11 caratteri.

Tramite i tasti numerici da 0 a 9 è possibile inserire i dati di input, inizialmente in metri quadrati, da un minimo di 0 (zero) a un massimo teorico di 999.999.999 (novecentonovantanovemilioni novecentonovantanovemila e novecentonovantanove) ogni volta. Il numero viene visualizzato in bianco, già ricomposto a gruppi di tre cifre per una migliore leggibilità, da destra verso sinistra (come per i display delle calcolatrici): è possibile comporre valori frazionari premendo il tasto del punto decimale (.).

La conferma per ogni inseri-

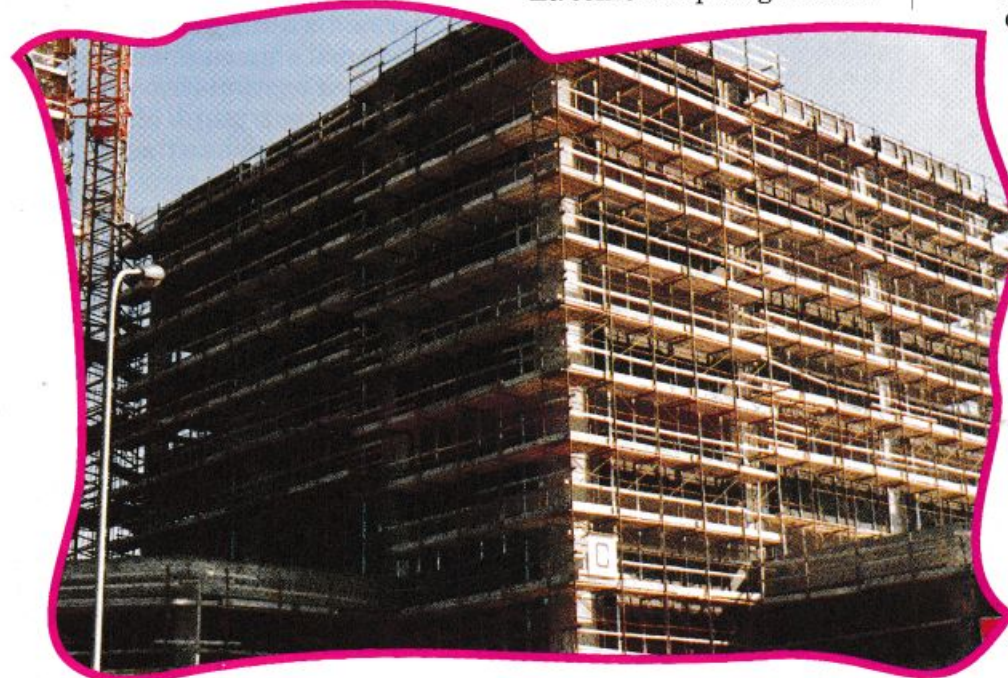
mento va data con il tasto Return, mentre eventuali correzioni sono sempre possibili, prima della conferma stessa, utilizzando il tasto della freccia sinistra (sul Commodore 64 è posto di fianco a quello numerico 1).

Premendo subito e solo il tasto Return alle richieste di dati si passa oltre (l'input sarà allora considerato come inesistente, cioè nullo). Anche inserendo uno zero o una serie di zeri si ottiene lo stesso effetto. Gli input nulli si danno quando per esempio la richiesta riguarda balconi, terrazze e simili, e l'immobile non ne ha.

Le richieste iniziali riguardano, si è detto, la superficie dell'immobile: si tratta di specificare fino a 10 diverse metrature, relativamente ai vani veri e propri della casa, alle autorimesse singole, ai posti macchina in uso comune, a balconi, terrazze, cantine e similari (distinguendo tra altezze utili di 1,70 metri e oltre oppure minori di questo limite), ad aree scoperte in godimento esclusivo (per esempio un cortile) e alla superficie condominiale a verde (metri quadri rapportati alla quota millimesimale spettante del totale).

I dati relativi ai vani veri e propri vanno scritti al netto di muri e parti perimetrali, cioè intendendo le aree effettivamente abitabili e utilizzabili: è consuetudine specificare al centimetro, cosicché per una stanza di 3,75 x 4,85 metri si inserisce il dato 18.1875 (metri quadri).

Il computer provvede a calcolare automaticamente i totali delle superfici di vani con altezza uguale e oltre 1,70 metri e minore di 1,70 metri, ricavando anche il totale della Superficie convenzionale, numero (già pesato con i coefficienti







previsti dalla legge) che permette poi di trovare, moltiplicato al Costo unitario di produzione, il Valore locativo.

Una seconda serie di input permette di quantificare proprio il costo unitario di produzione: per ogni richiesta è necessario scegliere, tramite i tasti funzione F1 e F3, la classificazione adeguata tra quelle disponibili nei vari elenchi, i quali sono tutti comodamente esplorabili a libro, avanti e indietro, fino a quando non si trova la voce esatta (da confermare con l'usuale tasto Return).

Dell'immobile si devono indicare, nell'ordine: la tipologia, come registrata a catasto; la classe demografica del comune di appartenenza, in base al numero di abitanti censiti; l'ubicazione, ricordando che la prima voce che compare, per edifici particolarmente degradati, va scelta solo se espressamente adottata dal comune di pertinenza; il livello di piano, ricordando che la specifica va fatta solo per gli edifici aventi almeno 3 piani fuori terra, e che l'agevolazione prevista dall'articolo 19 della legge (riduzione automatica del coefficiente di moltiplicazione e dunque dell'importo d'affitto) per livelli di piano oltre il terzo va adottata solo se non è disponibile l'ascensore; vetustà (da specificare con input numerico in anni), cioè periodo trascorso dalla finita costruzione o dall'ultimo restauro completo; stato di conservazione normale, mediocre o scadente, in base alle condizioni di pavimenti, pareti e soffitti, infissi, impianti elettrico, idrico (sanitari compresi) e di riscaldamento, elementi comuni come accessi, scale, ascensore, coperture e pareti; anno di ultimazione dei lavori e regione di appartenenza (del centro-nord o meridionale-insulare).

Da tutti i suddetti elementi il computer seleziona automatica-

mente la catena appropriata dei coefficienti di moltiplicazione e determina il preciso valore del costo unitario di produzione.

Il valore locativo viene ottenuto dalla moltiplicazione di superficie convenzionale per costo unitario di produzione: un ultimo input viene a questo punto richiesto ed è quello relativo alla facoltà data al locatore di calcolare il Canone annuo di base con un importo che può essere fino al 3,85% massimo del valore locativo determinato.

L'input va dato specificando la frazione corrispondente (0.0385 massimo): premendo invece subito e solo tasto Return, o inserendo un valore maggiore a quello consentito, si convalida comunque il 3.85% limite.

Una serie di sei videate mostra a questo punto, in sequenza, i vari risultati ottenuti, e in particolare l'ultimo è il canone annuale di base, cioè quello che il conduttore deve pagare per legge al locatore.

È importante ricordare che detta cifra va divisa in 12 importi uguali in caso di pagamenti mensili, ma soprattutto non va dimenticato, da ambo le parti contrattuali, che si tratta di Valore base, aumentabile e aggiornabile secondo quanto la legge



stessa prevede e consente. In particolare può essere aggiunto l'interesse legale sul capitale impiegato (cioè il 5 per cento) per la spesa di Riparazioni straordinarie importanti e prorogabili, così come è applicabile un aumento fino al 30% se l'unità immobiliare è già arredata completamente e adeguatamente.

Per quanto concerne invece l'aggiornamento a protezione dall'inflazione, si calcola ogni anno, in più, il 75% della variazione percentuale accertata dall'Istat e pubblicata periodicamente sulla Gazzetta Ufficiale.

Esempio: sui contratti di affitto stipulati al 1 agosto 1987, l'aumento applicabile è del 126.075%, cioè il 75% della variazione Istat sul periodo 01/08/78 - 01/08/87 (il 1978 è l'anno di riferimento per il calcolo). A tal pro-





**C64**

posito si fa presente che la variazione relativa al 1984 non va considerata, perché annullata a suo tempo da specifica legge.

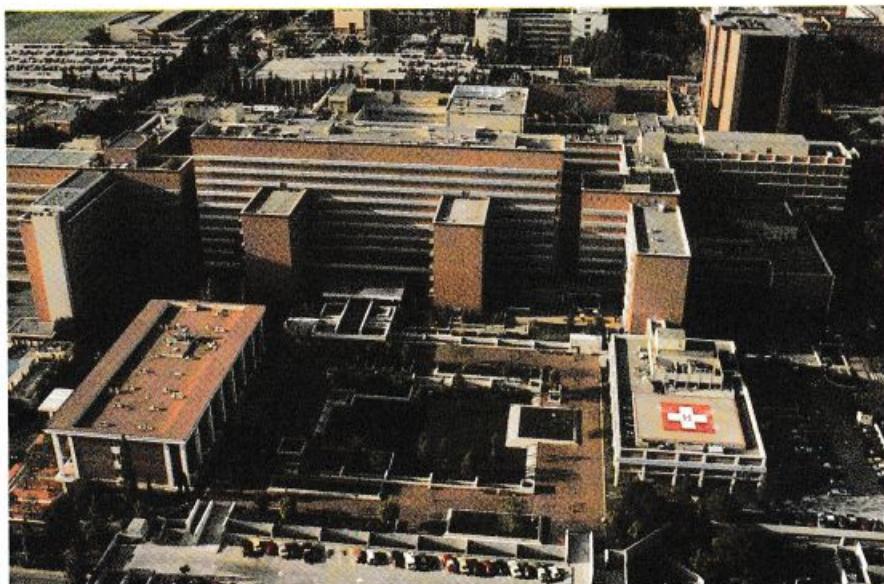
Concludendo, si può affermare che questo programma aiuta molto l'utente, effettuando tutte le procedure più noiose e complesse per il calcolo dell'equo canone.

La serie dei risultati ottenibili può servire per stendere i contratti di locazione ad uso abitativo con tranquillità e rapidità, e inoltre permette a chiunque di addentrarsi più agevolmente nel complesso mondo dei coefficienti moltiplicativi e delle tabelle tecniche di classificazione, che sono decine (e tutte già memorizzate nel computer).

### Un esempio

Vediamo ora un esempio di una tipica procedura di calcolo del canone annuale di base, anche al fine di mostrare come diventa facile ottenere risultati immediati e completi.

È il caso di un lussuoso attico sito in Milano zona Stazione Centrale, in un palazzo completamente restaurato nel 1978 e dotato di ascensore, con superficie



abitabile di mq 130.6 (altezza dei vani cm 240), più autorimessa singola di mq 23.45 (altezza cm 194), più cantina di mq 34 (altezza cm 165) più superficie scoperta in godimento esclusivo di mq 28, più quota condominiale a verde di mq 240.

Il totale superficie netta con vani di altezza cm 170 e oltre è di mq 422.05; il totale superficie netta con vani di altezza minore di cm 170 è di mq 34; il totale superficie convenzionale è di mq 176.475 (coefficienti di legge già

applicati).

L'immobile è censito come categoria A/1 (signorile), in zona demografica oltre i 400.000 abitanti (Milano e comune), ubicato tra periferia e centro, al livello di piano attico (senza riduzione di coefficiente come da Art. 19).

L'edificio ha 11 anni di vetustà, si trova in stato di conservazione normale, i lavori di restauro sono stati ultimati nell'anno 1978. La regione di riferimento è la Lombardia, quindi settentrionale.

Il costo unitario di produzione è di lire 1.201.996, dato dalla moltiplicazione tra coefficienti determinati e costo base al metro quadro, ovvero  $2 * 1.2 * 1.2 * 1.2 * 0.94 * 1 * 370.000$ .

Il valore locativo che se ne ricava è il prodotto tra superficie convenzionale e costo unitario di produzione, ovvero  $176.475 * 1.201.996$ , cioè lire 212.122.385.

Considerando infine un 3.85% applicabile, si ottiene il canone annuo di base, ossia lire 8.166.711, che possono diventare lire 18.462.891 al 1 agosto 1987, considerando anche il 75% della variazione Istat dal 1 agosto 1978 precedente. Insomma 12 rate mensili da lire 1.538.574 ca-dauna.

**Daniele Malavasi**

**EQUO**

(C) 1989 DISCOVOGUE

**CANONE 1**

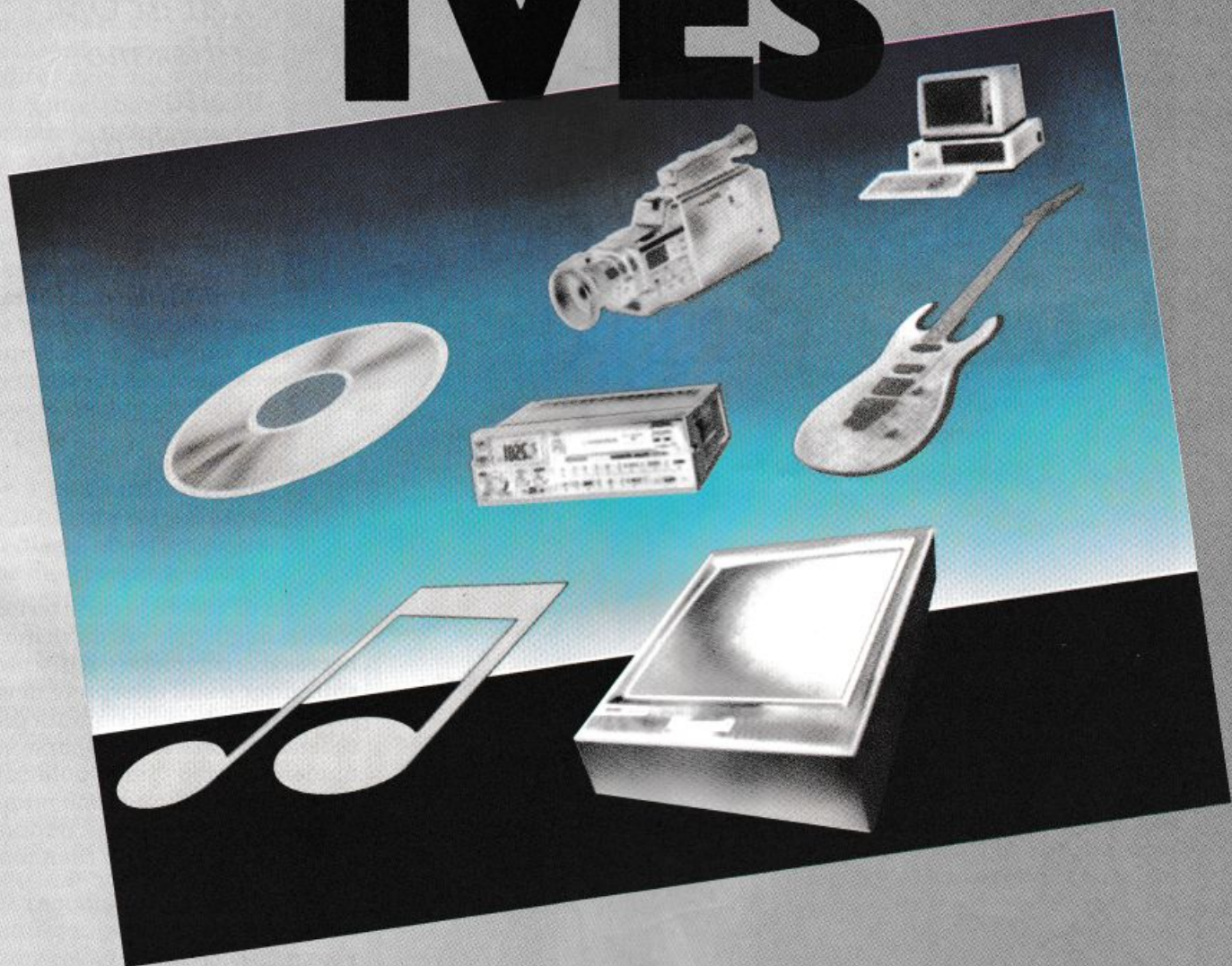
DETERMINAZIONE DEL CANONE ANNUO PER CONTRATTI DI LOCAZIONE AD USO ABITATIVO (RIF. L. 392/78 E MOD.)

SPECIFICARE SUPERFICIE CONDOMINIALE A VERDE CALCOLATA SULLA QUOTA SPETTANTE

> 240 <



# SIM-HI-FI IVES



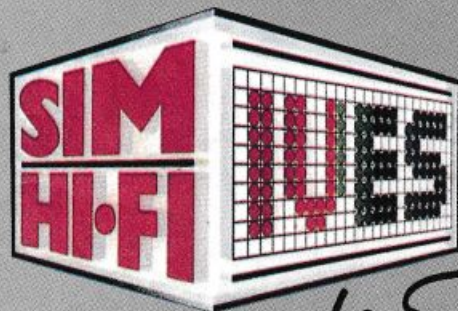
23° salone internazionale della musica e high fidelity  
international video and consumer electronics show

14-18

settembre 1989

Fiera Milano

STRUMENTI MUSICALI,  
ALTA FEDELTA', HOME VIDEO,  
HI-FI CAR, CAR ALARM SYSTEMS,  
PERSONAL COMPUTER, TV,  
VIDEOREGISTRAZIONE,  
ELETTRONICA DI CONSUMO.



*un grande Sim!*

Ingressi: Piazza Carlo Magno - Via Gattamelata - Orario: 9.00-18.00  
Aperto al pubblico: 14•15•16•17 - Giornata professionale: lunedì 18 settembre

ASSOEXPO

HOME  
VIDEO



Segreteria Generale SIM-HI-FI-IVES: Via Domenichino, 11 - 20149 Milano - Tel.: 02-4815541 - Telex: 313627 - Fax 02-4980330

VIVA  
i giovani





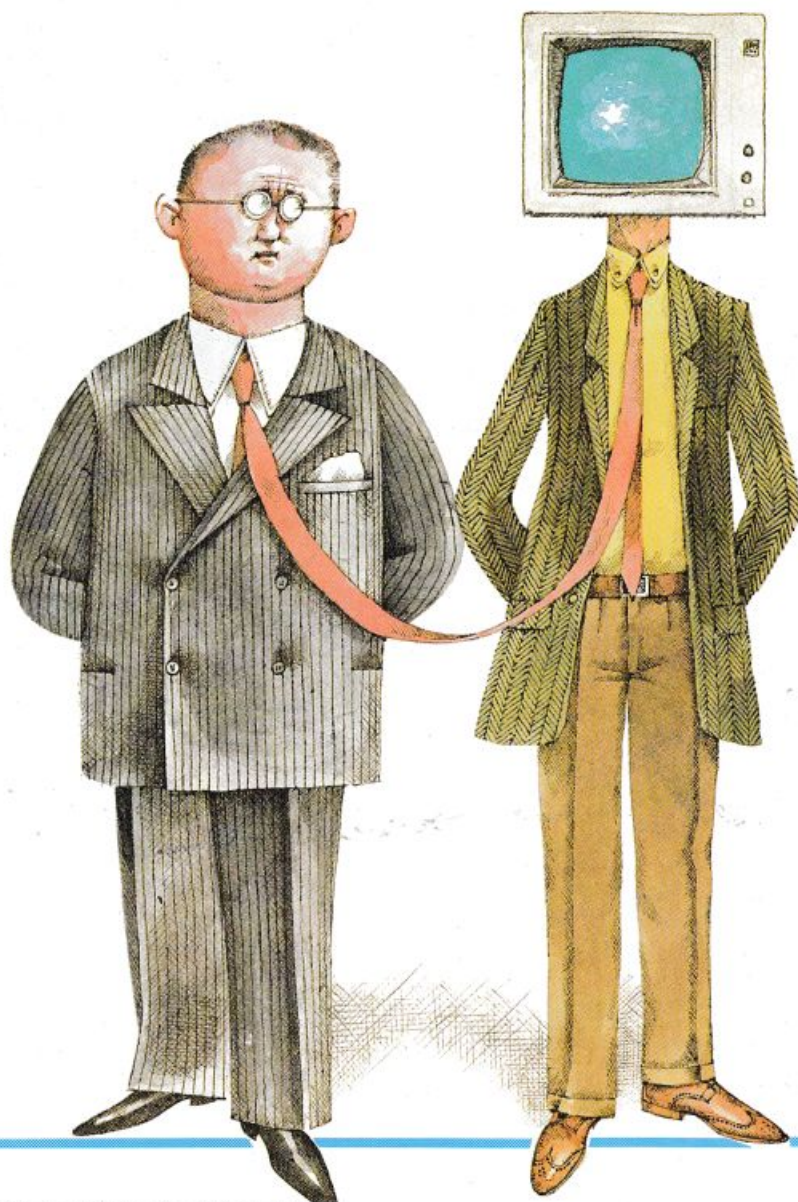


**C64E**

**SOFTWARE**

*L'argomento di questa puntata è lo Shell di Qpl, cioè l'interprete di comandi. Questo utilissimo strumento dell'ambiente integrato è molto potente, e mette a disposizione una miriade di comandi con cui praticamente potete fare tutto.*

# Come parlare all'interprete



Prima di vedere le caratteristiche del linguaggio più da vicino, è necessario completare il discorso sull'editor e sull'interprete di comandi iniziato nella puntata precedente. Il software comparso sul numero scorso sarà pubblicato nuovamente sul prossimo numero. Al termine di questa puntata sarete già in grado di compilare un programma

## **Lo Shell**

L'interprete di comandi, lo Shell, è ciò con cui siete in contatto quando attivate il Qpl. L'interprete è in grado di riconoscere e di eseguire tre tipi di comandi:

- **I comandi built in:** sono comandi predefiniti residenti in memoria o su disco. Due esempi di comandi di questo tipo sono Compile, che attiva il compilatore, e Map, che visualizza la mappa della memoria. Il primo di questi comandi, Compile, è su disco, mentre il secondo, Map, è residente in memoria.

- **I comandi definiti dall'utente e residenti in memo-**





**ria:** sono tutti i programmi compilati che l'utente ha inserito in memoria. Il codice oggetto prodotto dal compilatore viene automaticamente messo su disco. L'utente può però decidere di caricarlo in memoria, e così facendo aggiunge il programma stesso all'insieme di comandi disponibili. In altre parole, il programma compilato e caricato in memoria diventa un comando eseguibile dall'interprete.

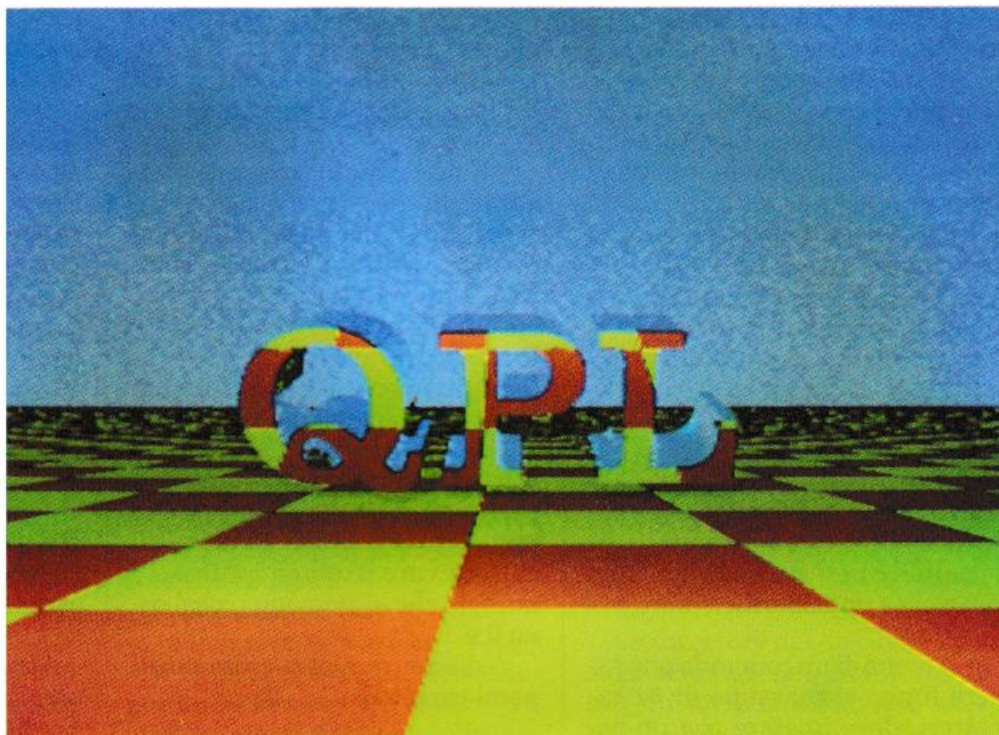
• **I comandi definiti dall'utente e residenti su disco:**

appartengono a questa categoria tutti i programmi compilati. Tutti i comandi residenti su disco (sia quelli built in sia quelli creati dall'utente) si riconoscono facilmente dal suffisso .c, che indica appunto la loro natura di file comando.

Per fare eseguire un comando all'interprete basta digitarne il nome e premere Return. Il nome del comando può essere scritto indifferentemente con lettere maiuscole o minuscole. Per scegliere il set maiuscolo o quello minuscolo dovete premere insieme il tasto Ctrl e il tasto della lettera a.

L'editor di linea dello Shell è molto comodo e dispone di numerosi comandi per facilitare l'inserimento e la rimozione di caratteri. Ecco l'elenco completo dei comandi di editing:

• **Return:** serve per comunicare all'interprete la fine della linea di comando. In pratica comunica all'interprete di eseguire il comando appena inserito. Non è indispensabile portare il cursore alla fine del comando prima di premere Return.



• **Del:** cancella il carattere alla sinistra del cursore senza trascinare la parte destra della stringa.

• **Ctrl seguito dalla freccia a sinistra (<):** cancella il carattere sotto il cursore trascinando la parte destra della stringa.

• **Crsr left e right:** spostano il cursore rispettivamente a sinistra e a destra.

• **Ctrl X:** cancella l'intera linea.

• **Ctrl K:** cancella la linea a partire dalla posizione del cursore verso destra.

• **Ctrl Y:** sposta il cursore alla fine della linea.

• **Ctrl J:** sposta il cursore all'inizio della linea.

• **Ctrl A:** attiva/disattiva il set maiuscolo.

• **Da F1 a F8:** cancella l'eventuale comando digitato e inserisce il comando associato al tasto funzione premuto. Per vedere la lista delle associazioni tasto-comando dovete usare il comando Help.

• **Ctrl B:** cancella l'eventuale

comando digitato e richiama l'ultimo comando inserito. Premendo nuovamente CTRL B si richiama il comando precedente a quello richiamato.

L'interprete ricorda gli ultimi 256 comandi inseriti, e quindi questo comando può essere usato per 256 volte. Premendo CTRL B, quando è visualizzato il primo comando della lista viene visualizzato l'ultimo comando inserito.

• **Ctrl Run/Stop:** interrompe immediatamente l'esecuzione del comando appena dato. Se il comando è residente su disco questo comando non interrompe il caricamento in memoria ma solo l'esecuzione.

• **Ctrl Z:** permette di inserire il carattere end of file. Il ruolo di questo comando sarà più chiaro quando verrà descritto il comando Type.

• **Run/Stop:** sospende temporaneamente l'esecuzione di un comando. L'esecuzione del comando riprende non appena il tasto viene rilasciato.





**C64**

Questi sono tutti i comandi di editing messi a disposizione dallo Shell e dovrebbero essere usati il più spesso possibile, perché permettono di velocizzare notevolmente l'inserimento dei comandi.

In particolare dovrete sempre tener presente i comandi Ctrl Run/Stop e Run/Stop, perché possono essere utili anche per il debugging dei programmi.

### **Caratteristiche dei comandi**

Prima di passare a descrivere in dettaglio i comandi built in disponibili è necessario fare alcune premesse importanti:

- Il nome di un comando può essere lungo al massimo di 14 caratteri, deve iniziare con un carattere e può contenere cifre e il trattino orizzontale basso ( \_ ), per ottenere questo carattere dovete premere il tasto freccia a sinistra.

Lo Shell accetta in input 80 caratteri, ma solo 14 al massimo so-

no quelli che possono costituire il nome del comando. Il motivo sta nel fatto che il DOS accetta nomi di 14 caratteri al massimo.

- Il nome di un comando residente su disco non necessariamente deve terminare col suffisso .c. Il suffisso .c viene automaticamente aggiunto se non viene specificato, ma se volete potete usare un suffisso diverso. Il suffisso può contenere un numero qualsiasi di caratteri.

- Il nome di un comando può essere preceduto da un prefisso numerico che indica il drive su cui ricercare il comando stesso. I numeri che si possono usare sono 0 e 1.

Vediamo qualche esempio di nomi corretti:

abc, pippo\_uno, base.dtp, 1:primo, 0:primo.

Gli ultimi due comandi vengono ricercati prima sul disco nel drive 1 e poi sul disco nel drive 0. Ecco invece alcuni esempi di no-

mi non corretti (tra parentesi è riportato il motivo per cui il nome è errato):

- 7primo (il nome di un comando non può iniziare con una cifra. Deve necessariamente cominciare con un carattere)

- uno.due.tre (il punto può essere usato solo una volta per identificare l'inizio dell'estensione del nome del comando)

- ecco\_un\_bel\_programma (il nome di un comando può contenere al massimo 14 caratteri).

### **Restrizioni**

Ci sono alcune restrizioni sulle estensioni che si possono utilizzare.

Alcune estensioni hanno infatti un significato particolare in Qpl. Ecco:

**.C:** indica un file comando.

**.D:** indica un file di dati.

**.E:** indica un file export. Il ruolo dei file export verrà spiegato nelle prossime puntate.

**.J:** indica un job file. I file di questo tipo contengono una serie di comandi che devono essere eseguiti in sequenza.

In pratica con i file job si possono creare delle vere e proprie procedure di comandi. Anche di questo riparleremo nelle prossime puntate.

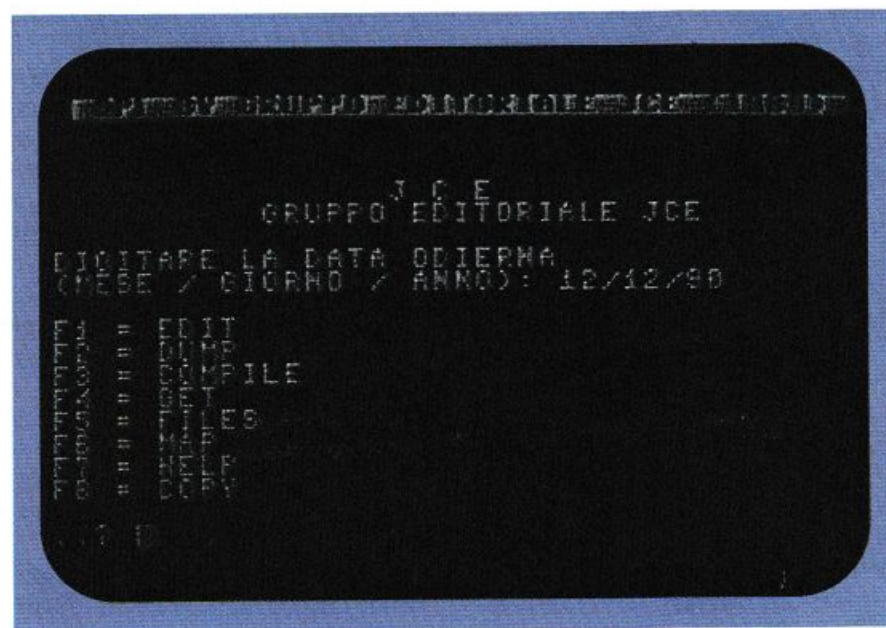
**.L:** indica un program listing. Anche questo concetto verrà chiarito in futuro.

**.R:** indica un file relativo. Il Qpl ha, tra l'altro, una serie di comode e potenti procedure per la gestione dei file relativi. Con queste procedure si possono realizzare database molto più facilmente che in Basic.

**.S:** indica un file sorgente preparato con l'editor del Qpl.

**.T:** indica un file testo creato con un editor diverso da quello del Qpl.

**.X:** indica un file contenente un



Un menù del programma che riassume i principali comandi comprensibili allo Shell di Qpl.





C64

cross reference. Anche di questo riparleremo nelle puntate successive.

Alcuni comandi richiedono dei valori numerici come argomenti. Tutti i valori numerici che devono essere passati ai comandi built in devono essere specificati in notazione esadecimale (senza il prefisso \$). I comandi creati dall'utente invece possono essere specificati indifferentemente in notazione esadecimale o decimale. Gli eventuali argomenti da passare a un comando vanno separati con lo spazio. Se volete invece includere uno o più spazi in un argomento dovete usare i doppi apici o l'apice singolo indifferentemente. Gli apici servono anche per inibire la conversione in formato minuscolo operata dall'interprete. Per esempio, nel comando:

```
set 5000 "impariamo il QPL"
```

(il comando set serve per porre in memoria una stringa), la stringa indicata fra doppi apici verrà passata come un unico argomento al comando set e i caratteri QPL non saranno tradotti in minuscolo. Invece nel comando:

```
set 5000 PIPPO
```

la stringa PIPPO viene automaticamente convertita in minuscolo.

### Le device

Lo Shell del Qpl, così come il linguaggio Qpl stesso, può effettuare operazioni di input e output su un certo numero di periferiche (device). Ogni periferica viene identificata in Qpl da un singolo carattere. Eccone l'elenco:

- **S:** lo schermo. È utilizzabile solo per l'output.

### Tavola 1.

#### Numeri e identificatori dei colori.

0 = black	8 = orange
1 = red	9 = brown
2 = white	a = ltred
3 = cyan	b = gray1
4 = purple	c = gray2
5 = green	d = ltgreen
6 = blue	e = ltblue
7 = yellow	f = gray3

- **K:** la tastiera. È utilizzabile solo per l'input.
- **P:** la stampante. È utilizzabile solo per l'output.
- **N:** il cestino. Serve, ovviamente, per scartare l'output.
- **W:** è un'area speciale della memoria, chiamata workspace. In pratica questa porzione della memoria funziona come RAM disk su cui si possono effettuare operazioni sia di input sia di output.
- **L:** è la libreria. È utilizzabile solo per l'input.
- **T:** il modem. È utilizzabile sia per l'input sia per l'output.

Gran parte dei comandi built in accettano uno di questi codici laddove si deve specificare il nome di un file. Per esempio, il comando type l visualizza il contenuto della libreria.

### Redirezione dell'I/O

Molti comandi del Qpl visualizzano qualcosa o prendono in input qualcosa.

Sia l'input sia l'output di un comando possono essere associati (rediretti) a un file particolare. L'utente, in pratica, può fare in modo che un comando mandi il suo output non sullo schermo ma su un file, oppure prenda l'input non dalla tastiera ma da un file. Per redirigere l'input e l'output si devono usare i seguenti caratteri:

```
< : redirige l'input
> : redirige l'output.
```

Facciamo qualche esempio. Il comando built in files visualizza il contenuto del disco nel drive. Se vogliamo memorizzare la directory in un file, piuttosto che vederla sullo schermo,

possiamo redirigere l'output del comando in questo modo:

```
files > disco_dir.t
```

La directory del disco nel drive sarà copiata nel file testo disco\_dir.t. Un modo utile di usare questo meccanismo consiste nel dirigere l'output di un comando sulla device .n. In questo modo si può eliminare completamente tutto l'output prodotto da un comando.

Consideriamo invece un ipotetico comando definito dall'utente che calcola la media di una certa serie di valori numerici inseriti alla tastiera. Chiamiamo questo comando media, e supponiamo di voler fare in modo che i valori da utilizzare per calcolare la media vengano letti da un file. Si può fare in questo modo:

```
media <dati.t
```

dove dati.t è il file testo contenente i dati da elaborare.

### Comandi built in

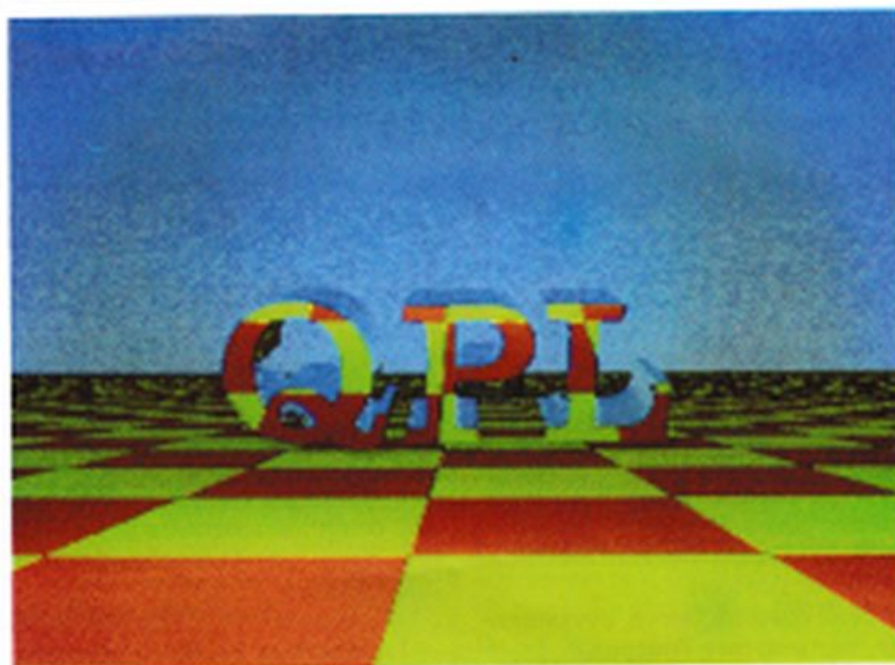
Ecco la descrizione dettagliata dei comandi built in disponibili:

- **Color:** permette di cambiare il colore di visualizzazione e del fondo dello schermo. La sintassi di questo comando è: color vis [sfondo]. Il primo parametro, ob-





C64E



bligatorio, indica il colore di di visualizzazione (cioè il colore del cursore) mentre il secondo, opzionale, indica il colore da assegnare allo sfondo dello schermo (le parentesi quadre non vanno specificate, servono solo per indicare che il parametro è opzionale). I due parametri possono essere dei numeri oppure degli identificatori predefiniti. La **tavola 1** riporta l'elenco dei valori e degli identificatori utilizzabili (accanto al simbolo di uguale viene indicato l'identificatore che può essere usato in luogo del numero):

- **Copy:** questo comando serve per copiare il contenuto di un file o di una device in un altro file o device. Vediamo qualche esempio:

- *copy myfile:* duplica il file myfile. L'interprete vi chiederà di cambiare disco prima di copiare il file specificato.
- *copy 0:uno 1:due:* copia il file uno del drive 0: nel file due del drive 1.
- *copy prog.s w:* copia il file prog.s nel workspace.
- *copy l s:* copia (visualizza) la li-

breria sullo schermo.

- *copy 1:pippo.sp:* copia il file pippo del drive 1: sulla stampante (stampa il file pippo).

Il comando Copy opera su un solo file alla volta e non consente l'uso di metacaratteri (come l'asterisco e il punto interrogativo). Nel caso in cui il file destinazione esista già, comparirà il messaggio:

append, replace, or cancel (a/r/c)?

premendo il tasto a il file sorgente viene accodato al file destinazione; premendo il tasto r il file già esistente viene cancellato; premendo il tasto c il comando Copy viene interrotto.

- **Cs:** cancella lo schermo e porta il cursore in alto a sinistra.

- **Date:** visualizza la data corrente e permette di modificarla. La data va inserita nel formato mm/gg/aa, cioè prima va inserito il mese, poi il giorno e infine l'anno. Ogni valore deve essere di due cifre.

- **Delete:** cancella uno o più file dal disco o dal workspace. La sintassi è: delete file1 file2... .Si possono cancellare più file contemporaneamente, ma non si possono usare metacaratteri nei nomi indicati (come per il comando Copy). Vediamo qualche esempio:

- delete myprog: cancella il file myprog.c (l'estensione .c viene messa automaticamente)
- delete w: cancella il workspace.

- **Dskcmd:** permette di inviare un comando al drive utilizzando la sintassi abituale. La sintassi di questo comando è: dskcmd "comando". Ecco un esempio:

dskcmd "n0:pippo,d2": formatta il disco nel drive.

- **Dyno:** visualizza lo stato del velocizzatore del drive, dynodisk, e permette di attivarlo/disattivarlo. Il velocizzatore duplica la velocità di lettura dei file e quindi dovreste usarlo sempre. Quando esso è attivo dovete assicurarvi che l'eventuale stampante o periferica collegata al bus seriale sia spenta, altrimenti potrebbe verificarsi il blocco del sistema. Il velocizzatore si attiva con Dyno on e si disattiva con Dyno off.

- **Dump:** visualizza il contenuto di una porzione di memoria in formato ASCII. La sintassi di questo comando è duplice: dump ind oppure dump inizio fine. Nel primo caso viene visualizzato il contenuto delle prime 8 locazioni a partire dall'indirizzo ind, mentre nel secondo caso viene visualizzato il contenuto di tutte le locazioni comprese fra gli indirizzi inizio e fine. Per bloccare momentaneamente il dump potete usare Run/Stop, mentre per terminare il comando dovete usare Ctrl Run/Stop.

Daniele Maggio





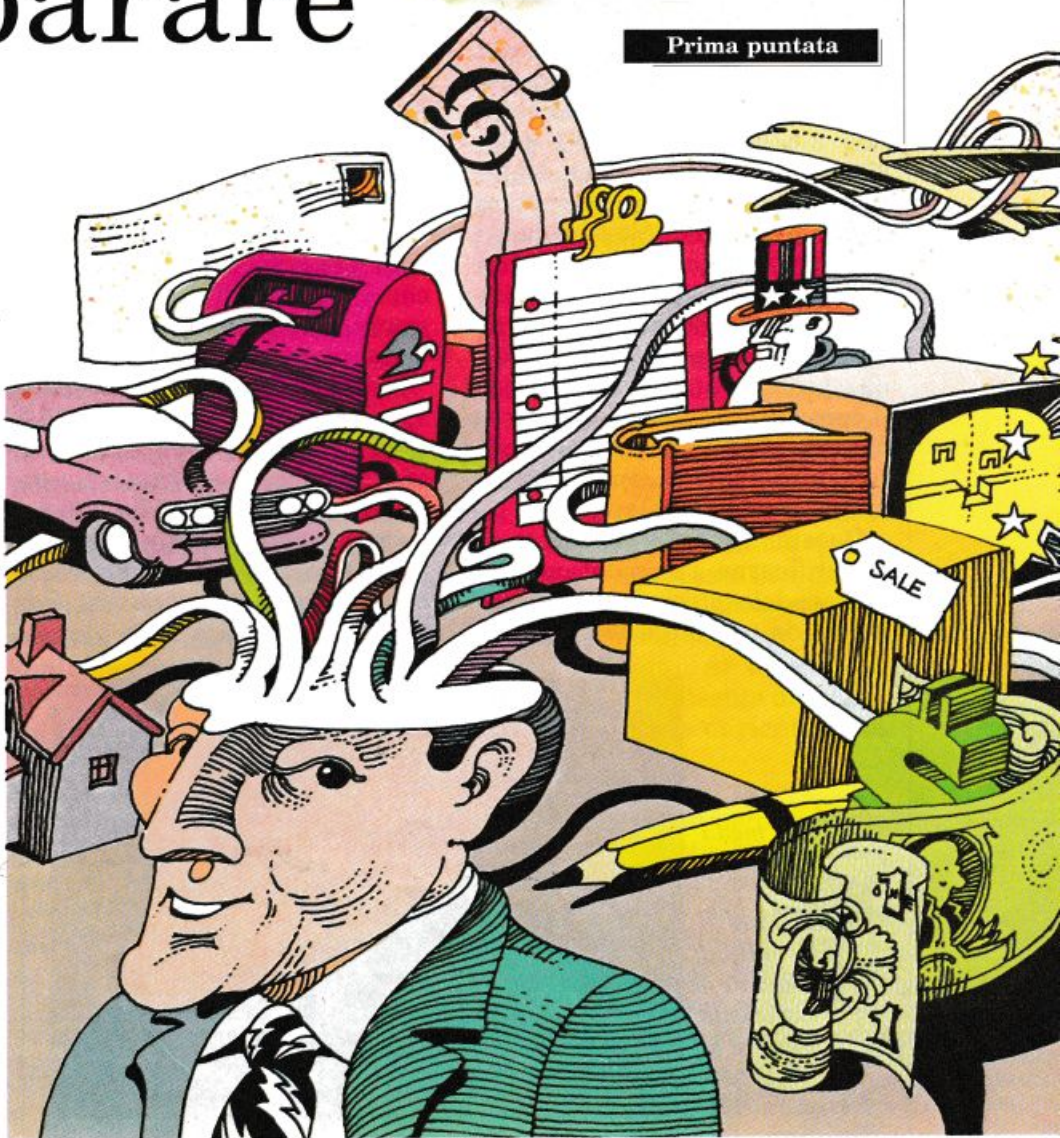
*Il programma che vi mettiamo a disposizione consente di effettuare elaborazioni di vario genere, spaziando da semplici e, per alcuni versi, divertenti calcoli, come la differenza percentuale, fino a elaborazioni più complesse di carattere commerciale, come il calcolo dell'interesse composto. Il tutto come esempio per chi impara a programmare.*

# Un buon esempio per imparare

Prima puntata

Chi volesse realizzare un programma di utilità varia dovrebbe necessariamente porsi il quesito: quali sono le necessità quotidiane che un computer può soddisfare?

Considerato che gli utenti di un computer (quale ne sia la potenza) tendono a demandare alla macchina molti, quando non troppi, aspetti della propria quotidianità, ne deriva che le possibilità sono praticamente infinite ed investono tutta la sfera del pensiero e della gestione sociale (dalla compilazione della schedina del totocalcio all'elaborazione di una dieta alimentare, alla probabilità che il primo premio della lotteria nazionale finisca a Bergamo piuttosto che a Velletri, da quanto percentualmente è aumentata la carne o lo zucchero, a quanto costerà il mutuo bancario). Ad alcuni di questi quesiti rispondono programmi creati appositamente, tipo quello del Totocalcio, del modello 740, della dieta alimentare, eccetera; noi abbiamo voluto creare un pacchetto che va dalla differenza percentuale (quanti sanno esattamente come formulare il con-







teggio?) alla quantificazione del montante di un mutuo, a tutta una serie di calcoli relativi alla soluzione di problemi di tempo, spazio e percorrenze per cui, tra il resto, saprete che percorrendo un chilometro in 32 secondi e 727 millesimi di secondo (a voi effettuare questi controlli manuali) non correrete il rischio di incorrere in qualche supermulta.

Passiamo perciò ad illustrare ognuna delle routine così come si propongono nel menù principale.

### Aliquote Iva

L'Iva è un problema che ormai interessa tutti e interferisce in ogni livello sociale. Non è più solo il problema dell'industriale, del commerciante o del professionista, ma anche dell'uomo della strada: anzi, quello che legalmente è definito il consumatore finale, è colui il quale deve fare i conti più meditati con questa imposta che, mentre per i settori indicati in precedenza è una partita di giro (tanta esce, tanta entra ed è previsto un conguaglio), per il consumatore finale è un costo che va a gravare irrimediabilmente su qualsiasi spesa che intenda, o debba, sostenere.

Questa routine consente di quantificarne l'ammontare, sia detraendolo (o, come si dice, risalendo) da un totale Iva inclusa,

sia aggiungendolo a un importo netto, o imponibile.

Non appena effettuata la selezione (che, come in tutti i menù e i sotto menù del programma, si ottiene utilizzando un solo carattere e cioè quello evidenziato), si presenta una maschera che propone la doppia opzione: Imponibile da totale e Totale da imponibile. A seconda che premiate la I piuttosto che la T cambierà lo schermo e vi verrà richiesto di inserire il dato opportuno: l'importo lordo di Iva piuttosto che quello imponibile. In entrambi i casi la domanda successiva si riferisce all'aliquota da utilizzare per il particolare settore della spesa che vi impegna. Al comparire dello sviluppo del conteggio, seguite le istruzioni che vengono impartite per arrivare a una opzione che vi consente di effettuare nuovamente un calcolo analogo, o di tornare al precedente sotto menù o al menù principale. Si precisa che l'uscita definitiva dal programma può essere effettuata solo dal menù principale.

Nel rispetto di una precisa disposizione di legge, nel corso del calcolo, l'ammontare dell'imposta viene sottoposto a un test condizionale e, nel caso vi fossero dei decimali, tale importo viene arrotondato alla lira superiore. È opportuno fare presente che, per legge, una tassa pagata male (e

per lo Stato la differenza di una lira è tassa pagata male) equivale a tassa non pagata, e pertanto tale cautela appare del tutto giustificata.

### Differenza percentuale

Chiedendo a bruciapelo quale sia, percentualmente, la differenza tra 80 e 100 (prescindendo da un eventuale significato concreto di tali numeri), nessuno stupore se ci sentissimo rispondere 20. In realtà è una di quelle domande trabocchetto del tipo: dimmi un colore, (rosso, di solito) piuttosto che: dimmi un numero. In realtà la risposta corretta è 25, mentre -20 (col segno meno) è quella tra 100 e 80. È pacifico che basta un minimo di riflessione, cioè evitare il bruciapelo, e la maggioranza fornirebbe la risposta corretta.

Poiché però, come diceva Ettore Petrolini, anche un solo spettatore è un pubblico, per lui è stata inserita questa routine che consente di valutare, col segno giusto e sino al sesto decimale, la differenza percentuale tra due grandezze (importi, misure, superfici, pesi, eccetera) semplicemente fornendo il dato di partenza e quello successivo.

Oltre al resto, non sempre un problema di questa natura presenta la ovvietà di un raffronto tra 80 e 100. Si sottolinea il fatto che valori che implicino dei decimali dovranno essere scritti alla maniera americana, separando la parte intera dalla decimale con un punto (.) e non con la virgola come siamo abituati a fare noi; la macchina segnalerebbe errore (Extra ignored), eseguendo comunque il conteggio, che sarebbe tuttavia inattendibile.

Come sempre, al termine viene proposta l'opzione che fornisce la possibilità di continuare altri confronti, o rientrare nel menù principale.

*Il menù principale. Per selezionare le opzioni desiderate è sufficiente premere la lettera corrispondente.*







### Equazione di secondo grado

In questa sede non ci si propone la soluzione di un problema di secondo grado, ma l'ottenimento dei due valori che definiscono l'equazione canonica:

$$ax^2+bx+c=0$$

(con valore algebrico per i segni "+")

dove a, b e c possono assumere valore di reali, in senso informatico (ossia non solo interi ma anche con decimali), ma non quello di espressioni. Se a vale 1/4, deve essere indicato con 0.25 (ricordare sempre il punto!). Analogamente per b e c.

Poiché lo sviluppo della formula prevede una doppia serie di calcoli in cui confluisce la soluzione di una radice quadrata, al cui interno compaiono, legati da relazioni aritmetiche, tutti gli indici, se si è incorsi in errore e il valore sotto radice assume segno negativo, il programma segnala tale errore e in luogo delle soluzioni compare la scritta *Espressione Errata*, offrendo la possibilità di ripetere l'operazione dopo il riesame dei calcoli a monte.

Per quanto superfluo si fa presente che non è possibile ottenere la radice quadrata di un numero negativo, perché nessun numero, elevato al quadrato, può assumere tale segno: questo lo sa anche la macchina.

Come anticipato, questa routine non vuole indurre gli studenti medi ad affidare al computer la soluzione di un problema di secondo grado: vuole solo sollevarli dalla fase finale che non presenta problemi di apprendimento, ma solo di banale, e per certi aspetti tediosa, finalizzazione di un elaborato che gli aspetti scolari deve averli risolti a monte, primo fra tutti quello della individuazione e della soluzione della logica del tema proposto.



*Il sottomenù delle aliquote Iva. Osservando il listato del programma è facile capirne la struttura e il funzionamento.*

Al termine del conteggio, il programma propone l'opzione per una eventuale ripetizione, o il rientro al menù principale.

### Fattura commerciale

Questa routine si riaggancia vagamente a quella del calcolo dell'Iva e prende il via dal totale imponibile cui si sarà giunti utilizzando a parte i dati elementari dei termini della fattura. Pertanto, una volta avviata con la pressione della lettera F, per prima cosa dovrà essere indicato tale valore, successivamente vengono richiesti le aliquote dell'Iva e della ritenuta d'acconto. Mentre la prima delle due dovrà sempre essere indicata, la seconda potrebbe non essere dovuta; in tal caso si potrà battere indifferentemente lo 0 (zero) o Return. Trattandosi di un dato numerico, per la macchina Return equivale a zero. Cambierà lo schermo e tutti i dati verranno indicati secondo l'opportuna collocazione. Inutile sottolineare che anche in questo caso viene effettuato il test condizionale per l'arrotondamento alla lira superiore nel caso in cui ciò si renda necessario.

### Interesse composto

Un capitale, investito per un certo periodo di tempo (di solito

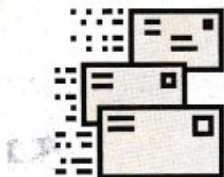
espresso in anni), a un determinato tasso d'interesse, produce al termine quello che viene chiamato montante e che rappresenta il capitale iniziale maggiorato degli interessi intermedi che a loro volta diventano capitale e così via sino allo spirare del periodo programmato. A titolo di esempio, un capitale investito al 7.16% circa, nell'arco di 10 anni si raddoppia, lira più, lira meno.

Una volta avviata la routine, il primo dato da inserire è quello del capitale iniziale, indi la durata dell'operazione che può anche essere espressa in termini frazionari; però attenzione: otto anni e sei mesi diventeranno una durata di 8.5. L'ultimo dato è quello del tasso d'interesse. Comparirà il montante, cioè il nuovo capitale disponibile al termine del periodo prefissato. Al termine la solita opzione consente di esaminare nuovi casi e di rientrare al menù principale.

Sul prossimo numero vedremo in dettaglio le altre tre funzioni del programma e effettueremo una breve analisi del listato per consentire a chiunque conosca, in termini generali, il linguaggio Basic di riprendere e eventualmente modificare le routine del programma.

**Giuseppe Alberti Monzoni**  
*(continua)*





## Diapason elettronico

Oltre ad avere un Commodore 64, sono un modesto, anche se appassionato, suonatore di chitarra. Essendo autodidatta ho qualche problema ad accordarla, specialmente quando debbo cambiare tutte le corde, il che, vivendo in una città di mare, capita con una certa frequenza. Il mio computer mi può aiutare? Ci sono dei manuali da consultare? Mi scuso per il problema che ai vostri livelli sembrerà poco interessante, ma a me risolverebbe qualcosa.

**Nunzio Salaino**  
Palermo

*Innanzitutto ci scusiamo per il ritardo con cui le rispondiamo, anche se, come al solito, la questione è stata esaminata immediatamente. C'era però il problema che, sino a qualche tempo fa, la frequenza del La (nota di 440 periodi) era lentamente salita per motivi "estetici" ma con danno di strumenti e di voci; proprio a seguito delle proteste dei professionisti del settore, è uscita una legge che ha ricondotto tale nota (il La) alla sua frequenza originale.*

*Era chiaro che, essendo le note musicali legate da un rapporto, la diversità di quella che ne è considerata il campione modificava anche tutte le altre, per cui avremmo potuto darle solo un'indicazione teorica, ma scostata da quella che era la pratica corrente; in concreto, le corde del suo strumento sarebbero state accordate tra di loro, ma tutte con una tonalità di poco più profonda, poiché il La che fornisce il Commodore 64 è proprio di 440 periodi.*

*Oggi, non solo possiamo dirle che il computer la può aiutare, ma sulla cassetta allegata alla ri-*

*vista è stato memorizzato un programma divertente per ottenere i suoni di ognuna delle corde vuote del suo strumento. Già che c'eravamo abbiamo inserito anche quelle del violino.*

*Non stiamo a indicare il funzionamento del programma anche perché, dopo la pressione di Run, appare una schermata che lo illustra e alla quale segue il primo di tre menù, il passaggio dall'uno all'altro dei quali potrà essere effettuato con la semplice pressione della lettera iniziale che lo individua.*

*Per quanto riguarda la seconda domanda che lei ci pone, per un utilizzo di questo tipo, è più che sufficiente il manuale che le è stato fornito con la macchina.*

*Tuttavia, se le sue ambizioni dovessero aumentare, ci scriva pure e le forniremo notizie più dettagliate sulle più recenti pubblicazioni in commercio.*

*Per leggere il listato del programma basta premere il tasto Run/Stop durante l'esecuzione dello stesso e digitare il comando List, oppure usare lo stesso comando appena terminato il caricamento del programma, prima di lanciarlo.*

*Sulla cassetta il programma si chiama Diapason e può essere facilmente caricato con il comando LOAD "DIAPASON" e la pressione del tasto Return.*

## Disco o cassetta?

Sono un vostro fedelissimo lettore da lungo tempo e, grazie agli interessantissimi argomenti trattati su *Radio Elettronica & Computer*, ho imparato a programmare molto bene. Utilissime le routine e la rubrica *Tips & Tricks*, fantastiche le utility,

simpatiche le recensioni e coinvolgenti i corsi di programmazione. Malgrado tutto questo non riesco a capire per quale ragione pubblicate spesso programmi che girano solo su disco quando la rivista esce solo su cassetta. Non sarebbe tutto più semplice se pubblicaste alcuni fascicoli allegati a una cassetta e altri su disco, secondo le percentuali di utenti-tape e di utenti-drive da voi stimate?

**Franco Cardano**  
Paderno Dugnano (MI)

*Gli utenti in possesso del registratore sono una percentuale molto bassa. Distribuire in modo uniforme nelle edicole di tutta Italia solo pochi esemplari di *Radio Elettronica & Computer* su cassetta è assai arduo, se non impossibile. La conseguenza di un tentativo simile sarebbe che gli utenti che possiedono solo il registratore a nastro non riuscirebbero più a trovare in edicola la rivista con la cassetta e, non potendo usare il dischetto, rimarrebbero completamente tagliati fuori.*

*La soluzione più democratica, almeno per ora, è quella che abbiamo adottato. Tutti i programmi sulla cassetta (a parte qualche videogame) possono essere trasferiti facilmente su disco.*

*Siccome un C64 senza drive limita anche la qualità dei programmi che supporta, e siccome è sempre difficile realizzare programmi adatti sia per drive che per tape, siamo costretti (piacevolmente per i cosiddetti utenti-drive) a pubblicare bei programmi solo per Drive 1541.*

*Per chi possiede solo il registratore, comunque, c'è sempre qualcosa di interessante su ogni numero.*



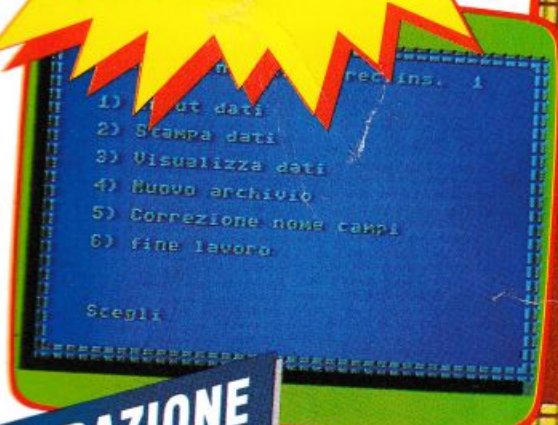
# Tutto **COMMODORE**

Anno III - Numero 25 - Luglio/Agosto 1989 - L.13.000

Gruppo Editoriale  
**JCE**

## Ufficio

**COPIATORE  
SINGLE-DRIVE PER  
40 TRACCE**



**FATTURAZIONE**

- Gestione archivi, stampa su due pagine, input controllato

**DATABASE**

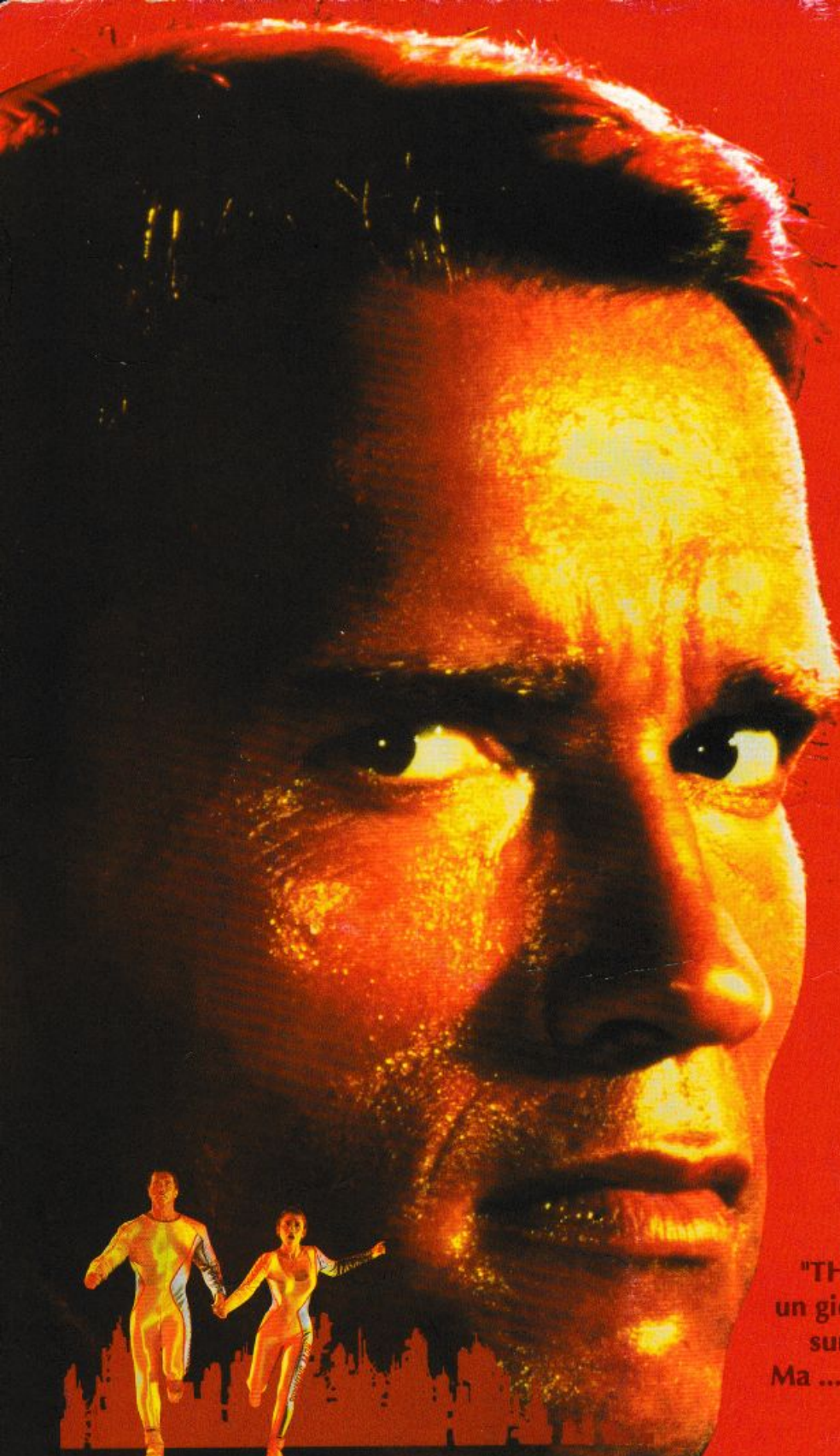
- Elasticità totale, stampa compatta, gestione a finestre



**è in edicola**



# SCHWARZENEGGER



AMIGA SHOTS

E' l'anno 2019  
"THE RUNNING MAN" è  
un gioco mortale a cui nes-  
suno è mai sopravvissuto.  
Ma ... SCHWARZENEGGER  
deve ancora giocare.

# THE RUNNING MAN

©1989 TAFT ENTERTAINMENT PICTURES/KEITH BARISH PRODUCTIONS

Distribuito in Italia da:  
LEADER Distribuzione  
Via Mazzini, 15  
21020 Casciago (VA)  
Tel. 0332/21 22 55



COMMODORE 64  
cass. / disco  
SPECTRUM cass.  
AMSTRAD cass.  
L. 15.000  
AMIGA  
L. 25.000  
ST/PC  
L. 29.000